

n8n Documentation

n8n GmbH

n8n Documentation

Welcome to n8n Docs

Where to start

About n8n

Join the community

Set up your n8n

Try it out

Structured Courses

Self-hosting n8n

Build a node

Stay updated

Choose your n8n

Platforms

Licenses

Free versions

Paid versions

Try it out

The very quick quickstart

Step one: Open a workflow template and sign up for n8n Cloud

Step two: Run the workflow

Step three: Add a node

Next steps

Your first workflow

Step one: Create a new workflow

Step two: Add a trigger node

Step three: Add the NASA node and set up credentials

Step four: Add logic with the If node

Step five: Output data from your workflow

Step six: Test the workflow

Congratulations

Next steps

Video courses

Beginner

Advanced

Text courses

Available courses

Level one: Introduction

Is this course right for me?

What will I learn in this course?

What do I need to get started?

How long does the course take?

How do I complete the course?

Navigating the Editor UI

Getting started

Editor UI settings

Nodes

Summary

Building a Mini-workflow

1. Add a Manual Trigger node

2. Add the Hacker News node

3. Configure the Hacker News node
 4. Execute the node
 5. Save the workflow
- Summary

Automating a (Real-world) Use Case

Understanding the scenario

Designing the Workflow

1. Getting data from the data warehouse

Create new workflow

Add an HTTP Request node

Get the data

What's next?

2. Inserting data into Airtable

Configure your table

Add an Airtable node to the HTTP Request node

Test the Airtable node

What's next?

3. Filtering Orders

Add If node before the Airtable node

Configure the If node

Insert data into Airtable

What's next?

4. Setting Values for Processing Orders

Add another node before the Airtable node

Configure the Edit Fields node

Add data to Airtable

What's next?

5. Calculating Booked Orders

About the Code node

Configure the Code node

What's next?

6. Notifying the Team

What's next?

7. Scheduling the Workflow

Remove the Manual Trigger node

Add the Schedule Trigger node

Connect the Schedule Trigger node

What's next?

8. Publishing and Examining the Workflow

Workflow Executions

Workflow Settings

What's next?

Exporting and importing workflows

Exporting and importing workflows

Test your knowledge

What's next?

Level two: Introduction

Is this course right for me?

What will I learn in this course?

What do I need to get started?

How long does the course take?

How do I complete the course?

Understanding the data structure

Data structure of n8n

Creating data sets with the Code node

Referencing node data with the Code node

Transforming data

Processing different data types

HTML and XML data

Date, time, and interval data

- Binary data
- Merging and splitting data
 - Merging data
 - Looping
 - Splitting data in batches
- Dealing with errors in workflows
 - Checking failed workflows
 - Catching erroring workflows
 - Throwing exceptions in workflows
- Automating a business workflow
 - Workflow design
 - Workflow prerequisites
- Workflow 1: Merging data
- Workflow 2: Generating reports
 - Part 1: Getting data from different sources
 - Part 2: Generating file for regional sales
 - Part 3: Generating files for total sales
- Workflow 3: Monitoring workflow errors
- Test your knowledge
 - What's next?
- Workflows
 - Create a workflow
 - Create a workflow
 - Run workflows manually
 - Run workflows automatically
 - Saving and publishing workflows
 - Checking publishing status
 - Publishing a workflow
 - Managing version history
 - Unpublishing a Workflow
 - Workflow components
 - Nodes
 - Add a node to your workflow
 - Node controls
 - Node settings
 - Connections
 - Create a connection
 - Delete a connection
 - Sticky Notes
 - Create a Sticky Note
 - Edit a Sticky Note
 - Change the color
 - Sticky Note positioning
 - Writing in Markdown
 - Make images full width
 - Embed a YouTube video
 - Executions
 - Execution modes
 - Execution lists
 - Manual, partial, and production executions
 - Manual executions
 - Partial executions
 - Production executions
 - Dirty nodes
 - How to recognize dirty node data
 - Why n8n marks nodes dirty
 - Resolving dirty nodes
 - Workflow-level executions list
 - View executions for a single workflow
 - Filter executions

- Retry failed workflows
- All executions
 - Filter executions
 - Retry failed workflows
 - Load data from previous executions into your current workflow
- Custom executions data
 - Set and access custom data using the Code node
- Debug and re-run past executions
 - Load data
- Tags
 - Add a tag to a workflow
 - Filter by tag
 - Manage tags
- Export and import workflows
 - Copy-Paste
 - From the Editor UI menu
 - From the command line
- Workflow templates
 - Access templates
 - Add your workflow to the n8n library
 - Self-hosted n8n: Use your own library
- Workflow sharing
 - Share a workflow
 - View shared workflows
 - Workflow roles and permissions
 - Node editing restrictions with unshared credentials
- Workflow settings
 - Access workflow settings
 - Available settings
- Streaming responses
 - Configure nodes for streaming
 - Important information
- Workflow history
 - Understand workflow history
 - View workflow history
 - Restore or copy previous versions
- Find your workflow ID
- Sub-workflow conversion
 - Selecting nodes for a sub-workflow
 - How to convert part of a workflow to a sub-workflow
 - Things to keep in mind
- Credentials
- Create and edit credentials
 - Create a credential
 - Expressions in credentials
- Credential sharing
 - Share a credential
 - Remove access to a credential
- User management
 - Setup guides
- Set up user management on n8n Cloud
 - Step one: In-app setup
 - Step two: Invite users
- Manage users
 - Delete a user
 - Resend an invitation to a pending user
- Account types
- Role-based access control (RBAC)
- RBAC role types
 - Project Admin

- Project Editor
- Project Viewer
- Create a project
 - Add and remove users in a project
 - Delete a project
 - Move workflows and credentials between projects or users
 - Using external secrets in projects
- Best practices for user management
 - All platforms
 - Self-hosted
- Two-factor authentication (2FA)
 - Enable 2FA
 - Disable 2FA for your instance
- Lightweight Directory Access Protocol (LDAP)
 - Enable LDAP
 - Merging n8n and LDAP accounts
 - LDAP user accounts in n8n
 - Turn LDAP off
- OpenID Connect (OIDC)
 - Set up OIDC
 - Setting up and enabling OIDC
 - Provider-specific OIDC setup
 - Discovery endpoints reference
 - Troubleshooting OIDC SSO
 - Known issues
 - Security Assertion Markup Language (SAML)
- Set up SAML
 - Enable SAML
 - Generic IdP setup
 - Setup resources for common IdPs
- Okta Workforce Identity SAML setup
 - Prerequisites
 - Setup
- Manage users with SAML
 - Exempt users from SAML
 - Deleting users
- Troubleshooting SAML SSO
- Keyboard shortcuts and controls
 - Workflow controls
 - Canvas
 - Node panel
 - Within nodes
 - Command bar
- Flow logic
 - Related sections
- Splitting workflows with conditional nodes
- Merging data
 - Merge data from different data streams
 - Merge data from different nodes
 - Merge data from multiple node executions
 - Compare, merge, and split again
- Looping in n8n
 - Using loops in n8n
 - Creating loops
 - Node exceptions
- Waiting
- Sub-workflows
 - Set up and use a sub-workflow
 - How data passes between workflows
 - Sub-workflow conversion

- Error handling
 - Create and set an error workflow
 - Error data
 - Cause a workflow execution failure using Stop And Error
- Execution order in multi-branch workflows
- Data
 - Related resources
- Data structure
 - Data item processing
- Data flow within nodes
- Transforming data
- Processing data with code
 - Function
- Data mapping
- Mapping in the UI
 - How to drag and drop data
- Mapping in the expressions editor
 - Access the linked item in a previous node's output
- Data item linking
- Item linking concepts
 - n8n's automatic item linking
 - Item linking example
- Item linking in the Code node
- Item linking errors
- Item linking for node creators
- Data pinning
 - Pin data
 - Unpin data
- Data editing
 - Edit output data
 - Use data from previous executions
- Data filtering
- Data mocking
 - Mocking with real data using data pinning
 - Generate custom data using the Code or Edit Fields nodes
 - Output a sample data set from the Customer Datastore node
- Data tables
 - Overview
 - How to use data tables
 - Considerations and limitations of data tables
 - Data tables versus variables
 - Exporting and importing data
- Binary data
 - Working with binary data in your workflows
 - Configure binary data mode when self-hosting
- Schema Preview
 - Using the preview
- n8n Cloud
- Cloud free trial
 - Upgrade to a paid account
 - Trial expiration
 - Enterprise trial
- Cloud admin dashboard
 - Access the dashboard from the app
 - Access the dashboard if the app is offline
- Update your Cloud version
 - Best practices for updating
 - Automatic update
- Set the Cloud instance timezone
- Cloud IP addresses

Cloud data management

- Memory limits on each Cloud plan
- How to reduce memory consumption in your workflow
- How to manage execution data on Cloud
- Cloud data pruning and out of memory incident prevention
- Change instance ownership
- Change instance username

Cloud concurrency

- Concurrency limits
- Details
- Comparison to queue mode

Download workflows

- How to download workflows
- Accessing workflows after your free trial

AI Assistant

- Current capabilities
- Tips for getting the most out of the Assistant
- FAQs

Source control and environments

Understand source control and environments

Environments in n8n

- Environments: What and why
- Environments in n8n

Git and n8n

- Git overview
- Branches: Multiple copies of a project
- Local and remote: Moving work between your machine and a Git provider
- Push, pull, and commit

Branch patterns

- Multiple instances, multiple branches
- Multiple instances, one branch
- One instance, multiple branches
- One instance, one branch

Set up source control for environments

- Prerequisites
- Step 1: Set up your repository and branches
- Step 2: Configure Git in n8n
- Step 3: Set up authentication
- Step 4: Connect n8n and configure your instance

Using source control and environments

Push and pull

- Fetch other people's work
- Send your work to Git
- What gets committed
- Merge behaviors and conflicts

Compare changes with workflow diffs

- Accessing workflow diffs
- Understanding the workflow diff view
- Reviewing node changes
- Viewing the summary of changes
- Navigating through each change
- Who can use workflow diffs

Copy work between environments

- Single branch
- Multiple branches
- Automatically send changes to n8n

Tutorial: Create environments with source control

- Choose your source control pattern
- Set up your repository

- Connect your n8n instances to your repository
- Push work from development
- Pull work to production
- Next steps
- External secrets
 - Connect n8n to your secrets store
 - Use secrets in n8n credentials
 - Using external secrets with n8n environments
 - Using external secrets in projects
 - Troubleshooting
- Log streaming
 - Set up log streaming
 - Events
 - Destinations
- Insights
 - Insights summary banner
 - Insights dashboard
 - Insights time periods
 - Setting the time saved by a workflow
 - Disable or configure insights metrics collection
 - Insights FAQs
- License Key
 - Add a license key using the UI
 - Add a license key using an environment variables
 - Allowlist the license server IP addresses
- Release notes
 - Semantic versioning in n8n
 - n8n@2.0.0
- Release notes pre 2.0
 - Semantic versioning in n8n
 - n8n@2.0.0-rc.4
 - n8n@1.123.3
 - n8n@2.0.0-rc.3
 - n8n@2.0.0-rc.1
 - n8n@1.123.2
 - n8n@2.0.0-rc.2
 - n8n@1.123.1
 - n8n@2.0.0-rc.0
 - n8n@1.123.1
 - n8n@1.123.0
 - n8n@1.122.5
 - n8n@1.122.4
 - n8n@1.122.3
 - n8n@1.122.2
 - n8n@1.122.0
 - n8n@1.121.3
 - n8n@1.121.2
 - n8n@1.121.1
 - n8n@1.121.0
 - n8n@1.120.4
 - n8n@1.120.2
 - n8n@1.120.1
 - n8n@1.120.0
 - n8n@1.119.2
 - n8n@1.119.1
 - n8n@1.118.2
 - n8n@1.119.0
 - n8n@1.118.1
 - n8n@1.117.3
 - n8n@1.118.0

n8n@1.117.2
n8n@1.117.1
n8n@1.117.0
n8n@1.116.2
n8n@1.115.4
n8n@1.116.1
n8n@1.116.0
n8n@1.116.0
n8n@1.115.3
n8n@1.115.2
n8n@1.114.4
n8n@1.115.0
n8n@1.114.3
n8n@1.114.2
n8n@1.114.1
n8n@1.114.0
n8n@1.113.3
n8n@1.112.6
n8n@1.113.2
n8n@1.112.5
n8n@1.113.1
n8n@1.112.4
n8n@1.113.0
n8n@1.112.3
n8n@1.111.1
n8n@1.110.2
n8n@1.112.2
n8n@1.112.1
n8n@1.112.0
n8n@1.111.0
n8n@1.110.1
n8n@1.109.2
n8n@1.110.0
n8n@1.109.1
n8n@1.108.2
n8n@1.109.0
n8n@1.107.4
n8n@1.108.1
n8n@1.107.3
n8n@1.108.0
n8n@1.107.2
n8n@1.107.1
n8n@1.106.3
n8n@1.107.0
n8n@1.106.2
n8n@1.106.1
n8n@1.105.4
n8n@1.105.3
n8n@1.106.0
n8n@1.105.2
n8n@1.105.1
n8n@1.104.2
n8n@1.105.0
n8n@1.104.1
n8n@1.103.2
n8n@1.104.0
n8n@1.101.3
n8n@1.102.4
n8n@1.103.1
n8n@1.102.3

n8n@1.103.0
n8n@1.94.0
n8n@1.93.0
n8n@1.92.2
n8n@1.91.3
n8n@1.92.1
n8n@1.92.0
n8n@1.91.2
n8n@1.90.3
n8n@1.91.1
n8n@1.91.0
n8n@1.90.2
n8n@1.90.1
n8n@1.90.0
n8n@1.89.2
n8n@1.89.1
n8n@1.89.0
n8n@1.88.0
n8n@1.87.2
n8n@1.86.1
n8n@1.87.1
n8n@1.87.0
n8n@1.86.0
n8n@1.85.4
n8n@1.84.3
n8n@1.84.2
n8n@1.85.3
n8n@1.85.2
n8n@1.85.1
n8n@1.85.0
n8n@1.84.1
n8n@1.84.0
n8n@1.83.2
n8n@1.82.4
n8n@1.82.3
n8n@1.83.1
n8n@1.83.0
n8n@1.82.2
n8n@1.82.1
n8n@1.82.0
n8n@1.81.4
n8n@1.81.3
n8n@1.81.2
n8n@1.80.5
n8n@1.80.4
n8n@1.81.1
n8n@1.81.0
n8n@1.80.3
n8n@1.79.4
n8n@1.80.2
n8n@1.79.3
n8n@1.80.1
n8n@1.79.2
n8n@1.80.0
n8n@1.75.3
n8n@1.74.4
n8n@1.79.1
n8n@1.78.1
n8n@1.77.4
n8n@1.76.4

n8n@1.79.0
n8n@1.77.3
n8n@1.78.0
n8n@1.77.2
n8n@1.76.3
n8n@1.77.1
n8n@1.76.2
n8n@1.77.0
n8n@1.76.1
n8n@1.76.0
n8n@1.75.2
n8n@1.74.3
n8n@1.75.1
n8n@1.74.2
n8n@1.75.0
n8n@1.74.1
n8n@1.74.0
n8n@1.73.1
n8n@1.73.0
n8n@1.72.1
n8n@1.71.3
n8n@1.72.0
n8n@1.71.2
n8n@1.70.4
n8n@1.71.1
n8n@1.70.3
n8n@1.71.0
n8n@1.70.2
n8n@1.70.1
n8n@1.70.0
n8n@1.68.1
n8n@1.69.2
n8n@1.69.1
n8n@1.69.0
n8n@1.68.0
n8n@1.67.1
n8n@1.67.0
n8n@1.66.0
n8n@1.65.2
n8n@1.64.3
n8n@1.65.1
n8n@1.65.0
n8n@1.64.2
n8n@1.64.1
n8n@1.64.0
n8n@1.63.4
n8n@1.62.6
n8n@1.63.3
n8n@1.63.2
n8n@1.62.5
n8n@1.63.1
n8n@1.62.4
n8n@1.63.0
n8n@1.62.3
n8n@1.62.2
n8n@1.62.1
n8n@1.61.0
n8n@1.59.4
n8n@1.60.1
n8n@1.60.0

n8n@1.59.3
n8n@1.59.2
n8n@1.59.1
n8n@1.58.2
n8n@1.59.0
n8n@1.58.1
n8n@1.58.0
n8n@1.57.0
n8n@1.56.2
n8n@1.56.1
n8n@1.56.0
n8n@1.55.3
n8n@1.55.2
n8n@1.55.1
n8n@1.54.4
n8n@1.54.3
n8n@1.54.2
n8n@1.55.0
n8n@1.54.1
n8n@1.53.2
n8n@1.54.0
n8n@1.45.2
n8n@1.53.1
n8n@1.53.0
n8n@1.52.2
n8n@1.52.1
n8n@1.51.2
n8n@1.52.0
n8n@1.51.1
n8n@1.50.2
n8n@1.51.0
n8n@1.50.1
n8n@1.50.0
n8n@1.49.0
n8n@1.48.3
n8n@1.47.3
n8n@1.48.2
n8n@1.47.2
n8n@1.48.1
n8n@1.48.0
n8n@1.47.1
n8n@1.47.0
n8n@1.45.1
n8n@1.46.0
n8n@1.44.2
n8n@1.42.2
n8n@1.45.0
n8n@1.44.1
n8n@1.44.0
n8n@1.43.1
n8n@1.43.0
n8n@1.42.1
n8n@1.41.1
n8n@1.42.0
n8n@1.41.0
n8n@1.40.0
n8n@1.39.1
n8n@1.38.2
n8n@1.37.4
n8n@1.39.0

n8n@1.38.1
n8n@1.37.3
n8n@1.38.0
n8n@1.37.2
n8n@1.36.4
n8n@1.36.3
n8n@1.37.1
n8n@1.36.2
n8n@1.37.0
n8n@1.36.1
n8n@1.36.0
n8n@1.35.0
n8n@1.34.2
n8n@1.34.1
n8n@1.34.0
n8n@1.33.1
n8n@1.32.2
n8n@1.33.0
n8n@1.32.1
n8n@1.31.2
n8n@1.32.0
n8n@1.31.1
n8n@1.31.0
n8n@1.30.1
n8n@1.30.0
n8n@1.29.1
n8n@1.29.0
n8n@1.27.3
n8n@1.28.0
n8n@1.27.2
n8n@1.27.1
n8n@1.27.0
n8n@1.26.0
n8n@1.25.1
n8n@1.25.0
n8n@1.24.1
n8n@1.22.6
n8n@1.24.0
n8n@1.22.5
n8n@1.23.0
n8n@1.22.4
n8n@1.22.3
n8n@1.22.2
n8n@1.22.1
n8n@1.22.0
n8n@1.18.4
n8n@1.21.1
n8n@1.18.3
n8n@1.21.0
n8n@1.20.0
n8n@1.19.5
n8n@1.18.2
n8n@1.19.4
n8n@1.19.0
n8n@1.18.1
n8n@1.18.0
n8n@1.17.1
n8n@1.17.0
n8n@1.16.0
n8n@1.15.2

n8n@1.15.1
n8n@1.14.2
n8n@1.14.1
n8n@1.14.0
n8n@1.13.0
n8n@1.12.2
n8n@1.12.1
n8n@1.11.2
n8n@1.12.0
n8n@1.11.1
n8n@1.11.0
n8n@1.10.1
n8n@1.9.3
n8n@1.9.2
n8n@1.10.0
n8n@1.9.1
LangChain in n8n (beta)
n8n@1.9.0
n8n@1.8.2
n8n@1.8.1
n8n@1.8.0
n8n@1.7.1
n8n@1.7.0
n8n@1.6.1
n8n@1.6.0
n8n@1.5.1
n8n@1.5.0
n8n@1.4.1
n8n@1.4.0
n8n@1.3.1
n8n@1.3.0
n8n@1.2.2
n8n@1.2.1
n8n@1.2.0
n8n@1.1.1
n8n@1.1.0
n8n@1.0.5
n8n@1.0.4
n8n@1.0.3
n8n@1.0.2
n8n@1.0.1

Release notes pre 1.0

Semantic versioning in n8n
n8n@0.237.0
n8n@0.236.3
n8n@0.236.2
n8n@0.236.1
n8n@0.236.0
n8n@0.234.1
n8n@0.235.0
n8n@0.234.0
n8n@0.233.1
n8n@0.233.0
n8n@0.232.0
n8n@0.231.3
n8n@0.231.2
n8n@0.231.1
n8n@0.231.0
n8n@0.230.3
n8n@0.230.2

n8n@0.230.1
n8n@0.230.0
n8n@0.229.0
n8n@0.228.2
n8n@0.228.1
n8n@0.228.0
n8n@0.227.1
n8n@0.227.0
n8n@0.226.2
n8n@0.226.1
n8n@0.226.0
n8n@0.225.2
n8n@0.225.1
n8n@0.225.0
n8n@0.224.4
n8n@0.224.2
n8n@0.224.1
n8n@0.224.0
n8n@0.223.0
n8n@0.222.3
n8n@0.222.2
n8n@0.222.1
n8n@0.222.0
n8n@0.221.3
n8n@0.221.2
n8n@0.221.1
n8n@0.221.0
n8n@0.220.1
n8n@0.220.0
n8n@0.219.1
n8n@0.219.0
n8n@0.218.0
n8n@0.217.2
n8n@0.217.1
n8n@0.217.0
n8n@0.216.3
n8n@0.216.2
n8n@0.215.4
n8n@0.215.3
n8n@0.214.5
n8n@0.214.4
n8n@0.216.1
n8n@0.216.0
n8n@0.215.2
n8n@0.215.1
n8n@0.215.0
n8n@0.214.3
n8n@0.214.2
n8n@0.214.1
n8n@0.214.0
n8n@0.213.0
n8n@0.212.1
n8n@0.212.0
n8n@0.211.2
n8n@0.211.1
n8n@0.211.0
n8n@0.210.2
n8n@0.210.1
n8n@0.210.0
n8n@0.209.4

n8n@0.209.3
n8n@0.209.2
n8n@0.209.1
n8n@0.209.0
n8n@0.208.1
n8n@0.208.0
n8n@0.207.1
n8n@0.207.0
n8n@0.206.1
n8n@0.206.0
n8n@0.205.0
n8n@0.204.0
n8n@0.203.1
n8n@0.203.0
n8n@0.202.1
n8n@0.202.0
n8n@0.201.0
n8n@0.200.1
n8n@0.200.0
n8n@0.199.0
n8n@0.198.2
n8n@0.198.1
n8n@0.198.0
n8n@0.197.1
n8n@0.197.0
n8n@0.196.0
n8n@0.195.5
n8n@0.195.4
n8n@0.195.3
n8n@0.195.2
n8n@0.195.1
n8n@0.195.0
n8n@0.194.0
n8n@0.193.5
n8n@0.193.4
n8n@0.193.3
n8n@0.193.2
n8n@0.193.1
n8n@0.193.0
n8n@0.192.2
n8n@0.192.1
n8n@0.192.0
n8n@0.191.1
n8n@0.191.0
n8n@0.190.0
n8n@0.189.1
n8n@0.189.0
n8n@0.188.0
n8n@0.187.2
n8n@0.187.1
n8n@0.187.0
n8n@0.186.1
n8n@0.186.0
n8n@0.185.0
n8n@0.184.0
n8n@0.183.0
n8n@0.182.1
n8n@0.182.0
n8n@0.181.2
n8n@0.181.1

n8n@0.181.0
n8n@0.180.0
n8n@0.179.0
n8n@0.178.2
n8n@0.178.1
n8n@0.178.0
n8n@0.177.0
n8n@0.176.0
n8n@0.175.1
n8n@0.175.0
n8n@0.174.0
n8n@0.173.1
n8n@0.173.0
n8n@0.172.0
n8n@0.171.1
n8n@0.171.0
n8n@0.170.0
n8n@0.169.0
n8n@0.168.2
n8n@0.168.1
n8n@0.168.0
n8n@0.167.0
n8n@0.166.0
n8n@0.165.0
n8n@0.164.1
n8n@0.164.0
n8n@0.163.1
n8n@0.163.0
n8n@0.162.0
n8n@0.161.1
n8n@0.161.0
n8n@0.160.0
n8n@0.159.1
n8n@0.159.0
n8n@0.158.0
n8n@0.157.1
n8n@0.157.0
n8n@0.156.0
n8n@0.155.2
n8n@0.154.0
n8n@0.153.0
n8n@0.152.0
n8n@0.151.0
n8n@0.150.0
n8n@0.149.0
n8n@0.148.0
n8n@0.147.1
n8n@0.147.0
n8n@0.146.0
n8n@0.145.0
n8n@0.144.0
n8n@0.143.0
n8n@0.142.0
n8n@0.141.1
n8n@0.141.0
n8n@0.140.0
n8n@0.139.1
n8n@0.139.0
n8n@0.138.0
n8n@0.137.0

n8n@0.136.0
n8n@0.135.3
n8n@0.135.2
n8n@0.135.1
n8n@0.135.0
n8n@0.134.0
n8n@0.133.0
n8n@0.132.2
n8n@0.132.1
n8n@0.132.0
n8n@0.131.0
n8n@0.130.0
n8n@0.129.0
n8n@0.128.0
n8n@0.127.0
n8n@0.126.1
n8n@0.126.0
n8n@0.125.0
n8n@0.124.1
n8n@0.124.0
n8n@0.123.1
n8n@0.123.0
n8n@0.122.3
n8n@0.122.2
n8n@0.122.1
n8n@0.122.0
n8n@0.121.1
n8n@0.121.0
n8n@0.120.0
n8n@0.119.0
n8n@0.118.1
n8n@0.118.0
n8n@0.117.0
n8n@0.116.1
n8n@0.116.0
n8n@0.115.0
n8n@0.114.0
n8n@0.113.0
n8n@0.112.0
n8n@0.111.0
n8n@0.110.3
n8n@0.110.0
n8n@0.109.0
n8n@0.108.0
n8n@0.107.0
n8n@0.106.0
n8n@0.105.0
n8n@0.104.2
n8n@0.104.1
n8n@0.104.0
n8n@0.103.1
n8n@0.103.0
n8n@0.102.0
n8n@0.101.0
n8n@0.100.0
n8n@0.99.1
n8n@0.99.0
n8n@0.98.0
n8n@0.96.0
n8n@0.95.1

n8n@0.95.0
n8n@0.94.1
n8n@0.94.0
n8n@0.93.0
n8n@0.92.0
n8n@0.91.0
n8n@0.90.0
n8n@0.89.2
n8n@0.89.1
n8n@0.89.0
n8n@0.88.1
n8n@0.88.0
n8n@0.87.2
n8n@0.87.1
n8n@0.87.0
n8n@0.86.1
n8n@0.86.0
n8n@0.85.0
n8n@0.84.4
n8n@0.84.3
n8n@0.84.2
n8n@0.84.1
n8n@0.84.0
n8n@0.83.0
n8n@0.82.1
n8n@0.82.0
n8n@0.81.0
n8n@0.80.0
n8n@0.79.3
n8n@0.79.2
n8n@0.79.1
n8n@0.79.0
n8n@0.78.0
n8n@0.76.0
n8n@0.75.0
n8n@0.74.0
n8n@0.73.1
n8n@0.73.0
n8n@0.72.0
n8n@0.71.0
n8n@0.70.0

n8n v2.0 breaking changes

- Behavior changes
- Security
- Data
- Configuration & Environment
- CLI & Workflow
- External Hooks
- Release channels
- Reporting issues

n8n v2.0 migration tool

- Accessing the Tool
- Understanding Your Migration Status
- Viewing Issues
- Understanding Empty States
- Recommended Workflow

n8n v1.0 migration guide

- New features
- Deprecations
- Breaking changes

- Updating to n8n 1.0
- Reporting issues
- Thank you
- Get help with n8n
 - Where to get help
 - What to include in your message
- How can you contribute?
 - Share some love: Review us
 - Help out the community
 - Contribute a workflow template
 - Build a node
 - Contribute to the code
 - Contribute to the docs
 - Contribute to community tutorials
 - Refer a candidate
- Privacy and security at n8n
- Privacy
 - GDPR
 - Data collection
 - Retention and deletion of personal identifiable data
 - Payment processor
- Incident response
- What you can do
 - All users
 - Self-hosted users
- Sustainable Use License
- License FAQs
- Integrations
 - Built-in nodes
 - Community nodes
 - Credential-only nodes and custom operations
 - Generic integrations
 - Where to go next
- Built-in integrations
 - Core nodes
 - Cluster nodes
 - Credentials
 - Community nodes
- Core nodes library
- Activation Trigger node
 - Node parameters
 - Templates and examples
- Aggregate
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- AI Transform
 - Node parameters
 - Templates and examples
- Code node
 - Common issues
- Keyboard shortcuts when using the Code editor
 - Cursor Movement
 - Selection
 - Basic Operations
 - Delete Operations
 - Line Operations
 - Autocomplete
 - Indentation

- Code Folding
- Multi-cursor
- Formatting
- Search & Navigation
- Code node common issues
 - Code doesn't return items properly
 - A 'json' property isn't an object
 - Code doesn't return an object
 - 'import' and 'export' may only appear at the top level
 - Cannot find module '<module>'
 - Using global variables
- Compare Datasets
 - Node parameters
 - Understand item comparison
 - Node options
 - Understand the output
 - Templates and examples
- Compression
 - Node parameters
 - Templates and examples
- Chat Trigger node
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
 - Set the chat response manually
 - Common issues
- Chat Trigger node common issues
 - Pass data from a website to an embedded Chat Trigger node
 - Chat Trigger node doesn't fetch previous messages
- Convert to File
 - Operations
 - Templates and examples
- Crypto
 - Actions
 - Node parameters
 - Templates and examples
- Data table
 - Node parameters
 - Related resources
- Date & Time
 - Operations
 - Add to a Date
 - Extract Part of a Date
 - Format a Date
 - Get Current Date
 - Get Time Between Dates
 - Round a Date
 - Subtract From a Date
 - Templates and examples
 - Related resources
- Debug Helper
 - Operations
 - Node parameters
 - Templates and examples
- Edit Fields (Set)
 - Node parameters
 - Node options
 - Templates and examples
 - Arrays and expressions in JSON Output mode

- [Edit Image](#)
 - [Operations](#)
 - [Node parameters](#)
 - [Node options](#)
 - [Templates and examples](#)
- [Email Trigger \(IMAP\) node](#)
 - [Operations](#)
 - [Node parameters](#)
 - [Node options](#)
 - [Templates and examples](#)
- [Error Trigger node](#)
 - [Usage](#)
 - [Templates and examples](#)
 - [Related resources](#)
 - [Error data](#)
- [Evaluation node](#)
 - [Operations](#)
 - [Templates and examples](#)
 - [Related resources](#)
- [Evaluation Trigger node](#)
 - [Parameters](#)
 - [Templates and examples](#)
 - [Related resources](#)
- [Execute Command](#)
 - [Node parameters](#)
 - [Templates and examples](#)
 - [Common issues](#)
- [Execute Command node common issues](#)
 - [Command failed: <command> /bin/sh: <command>: not found](#)
 - [Error: stdout maxBuffer length exceeded](#)
- [Execute Sub-workflow](#)
 - [Node parameters](#)
 - [Node options](#)
 - [Templates and examples](#)
 - [Set up and use a sub-workflow](#)
 - [How data passes between workflows](#)
- [Execute Sub-workflow Trigger node](#)
 - [Usage](#)
 - [Templates and examples](#)
 - [How data passes between workflows](#)
- [Execution Data](#)
 - [Operations](#)
 - [Data to Save](#)
 - [Limitations](#)
 - [Templates and examples](#)
- [Extract From File](#)
 - [Operations](#)
 - [Example workflow](#)
 - [Node parameters](#)
 - [Templates and examples](#)
- [Filter](#)
 - [Node parameters](#)
 - [Node options](#)
 - [Templates and examples](#)
- [FTP](#)
 - [Operations](#)
 - [Delete](#)
 - [Download](#)
 - [List](#)
 - [Rename](#)

- Upload
- Templates and examples
- Git
 - Operations
 - Add
 - Add Config
 - Clone
 - Commit
 - Fetch
 - List Config
 - Log
 - Pull
 - Push
 - Push Tags
 - Status
 - Tag
 - User Setup
 - Templates and examples
- GraphQL
 - Node parameters
 - Headers
 - Templates and examples
 - Related resources
- Guardrails node
 - Node parameters
- HTML
 - Operations
 - Generate HTML template
 - Extract HTML Content
 - Convert to HTML Table
 - Templates and examples
- HTTP Request node
 - Node parameters
 - Node options
 - Tool-only options
 - Import curl command
 - Templates and examples
 - Common issues
- HTTP Request node common issues
 - Bad request - please check your parameters
 - The resource you are requesting could not be found
 - JSON parameter need to be an valid JSON
 - Forbidden - perhaps check your credentials
 - 429 - The service is receiving too many requests from you
- If
 - Add conditions
 - Templates and examples
 - Branch execution with If and Merge nodes
 - Related resources
- JWT
 - Operations
 - Node parameters
 - Payload Claims
 - Node options
 - Templates and examples
- LDAP
 - Operations
 - Compare
 - Create
 - Delete

- Rename
- Search
- Update
- Templates and examples
- Limit
 - Node parameters
 - Templates and examples
 - Related resources
- Local File Trigger node
 - Node parameters
 - Changes to a Specific File
 - Changes Involving a Specific Folder
 - Node options
 - Templates and examples
- Loop Over Items
 - When to use the Loop Over Items node
 - Node parameters
 - Node options
 - Templates and examples
- Manual Trigger node
 - Common issues
- Markdown
 - Operations
 - Node parameters
 - Node options
 - Templates and examples
 - Parsers
- MCP Client node
 - Node parameters
 - Options
 - Templates and examples
 - Related resources
- MCP Server Trigger node
 - How the MCP Server Trigger node works
 - Node parameters
 - Templates and examples
 - Limitations
 - Related resources
 - Common issues
- Merge
 - Node parameters
 - Templates and examples
 - Merging data streams with uneven numbers of items
 - Branch execution with If and Merge nodes
 - Try it out: A step by step example
- n8n
 - Operations
 - Generate audit
 - Create credential
 - Delete credential
 - Get credential schema
 - Get execution
 - Get many executions
 - Delete execution
 - Activate, deactivate, delete, and get workflow
 - Create workflow
 - Get many workflows
 - Update workflow
 - Templates and examples
- n8n Form node

- Setting up the node
- Running the node
- Templates and examples
- n8n Form Trigger node
 - Build and test workflows
 - Production workflows
 - Set default selections with query parameters
 - Node parameters
 - Node options
 - Customizing Form Trigger node behavior
 - Templates and examples
- n8n Trigger node
 - Node parameters
 - Templates and examples
- No Operation, do nothing
 - Templates and examples
- Read/Write Files from Disk
 - Operations
 - Read File(s) From Disk
 - Write File to Disk
 - Templates and examples
 - File locations
- Remove Duplicates node
 - Operation modes
 - Templates and examples
 - Related resources
- Templates and examples
 - Templates
 - Set up sample data using the Code node
 - Removing duplicates from the current input
 - Keep items where the value is new
 - Keep items where the value is higher than any previous value
 - Keep items where the value is a date later than any previous date
- Rename Keys
 - Node parameters
 - Node options
 - Templates and examples
- Respond to Chat node
 - Node parameters
 - Node options
 - Related resources
 - Common issues
- Respond to Webhook
 - How to use Respond to Webhook
 - Node parameters
 - Node options
 - How n8n secures HTML responses
 - Templates and examples
 - Workflow behavior
 - Output the response sent to the webhook
 - Return more than one data item (deprecated)
- RSS Read
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- RSS Feed Trigger node
 - Node parameters
 - Templates and examples
 - Related resources

- Schedule Trigger node
 - Node parameters
 - Templates and examples
 - Common issues
- Schedule Trigger node common issues
 - Invalid cron expression
 - Scheduled workflows run at the wrong time
- Send Email
 - Node parameters
 - Node options
 - Waiting for a response
 - Templates and examples
- Sort
 - Node parameters
 - Templates and examples
 - Related resources
- Split Out
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- SSE Trigger node
 - Node parameters
 - Templates and examples
- SSH
 - Operations
 - Templates and examples
- Stop And Error
 - Operations
 - Node parameters
 - Templates and examples
 - Related resources
- Summarize
 - Node parameters
 - Node options
 - Ignore items without valid fields to group by
 - Templates and examples
 - Related resources
- Switch
 - Node parameters
 - Templates and examples
 - Related resources
- TOTP
 - Node parameters
 - Node options
 - Templates and examples
- Wait
 - Operations
 - Templates and examples
 - Time-based operations
- Webhook node
 - Workflow development process
 - Node parameters
 - Node options
 - How n8n secures HTML responses
 - Templates and examples
 - Common issues
- Workflow development
 - Build and test workflows
 - Production workflows

Common issues and questions

- Listen for multiple HTTP methods
- Use the HTTP Request node to trigger the Webhook node
- Use curl to trigger the Webhook node
- Send a response of type string
- Test URL versus Production URL
- IP addresses in whitelist are failing to connect
- Only one webhook per path and method
- Timeouts on n8n Cloud
- Workflow Trigger node
 - Node parameters
 - Templates and examples
- XML
 - Node parameters
 - Node options
 - Templates and examples
- Actions library
- Action Network node
 - Operations
 - Templates and examples
- ActiveCampaign node
 - Operations
 - Templates and examples
- Adalo node
 - Operations
 - Templates and examples
 - Related resources
- Affinity node
 - Operations
 - Templates and examples
- Agile CRM node
 - Operations
 - Templates and examples
- Airtable node
 - Operations
 - Templates and examples
 - Related resources
 - Node reference
 - Common issues
- Airtable node common issues
 - Forbidden - perhaps check your credentials
 - Service is receiving too many requests from you
- Airtop node
 - Operations
 - Templates and examples
 - Related resources
 - Node reference
- AMQP Sender node
 - Operations
 - Templates and examples
- Anthropic node
 - Operations
 - Templates and examples
 - Related resources
- APITemplate.io node
 - Operations
 - Templates and examples
- Asana node
 - Operations
 - Templates and examples

- Autopilot node
 - Operations
 - Templates and examples
- AWS Certificate Manager node
 - Operations
 - Templates and examples
 - Related resources
- AWS Cognito node
 - Operations
 - Templates and examples
 - Related resources
- AWS Comprehend node
 - Operations
 - Templates and examples
- AWS DynamoDB node
 - Operations
 - Templates and examples
- AWS Elastic Load Balancing node
 - Operations
 - Templates and examples
 - Related resources
- AWS IAM node
 - Operations
 - Templates and examples
 - Related resources
- AWS Lambda node
 - Operations
 - Templates and examples
- AWS Rekognition node
 - Operations
 - Templates and examples
- AWS S3 node
 - Operations
 - Templates and examples
- AWS SES node
 - Operations
 - Templates and examples
- AWS SNS node
 - Operations
 - Templates and examples
- AWS SQS node
 - Operations
 - Templates and examples
- AWS Textract node
 - Operations
 - Templates and examples
- AWS Transcribe node
 - Operations
 - Templates and examples
- Azure Cosmos DB node
 - Operations
 - Templates and examples
 - Related resources
- Azure Storage node
 - Operations
 - Templates and examples
 - Related resources
- BambooHR node
 - Operations
 - Templates and examples

- Bannerbear node
 - Operations
 - Templates and examples
- Baserow node
 - Operations
 - Templates and examples
- Beeminder node
 - Operations
 - Templates and examples
- Bitly node
 - Operations
 - Templates and examples
- Bitwarden node
 - Operations
 - Templates and examples
- Box node
 - Operations
 - Templates and examples
- Brandfetch node
 - Operations
 - Templates and examples
- Brevo node
 - Operations
 - Templates and examples
- Bubble node
 - Operations
 - Templates and examples
- Chargebee node
 - Operations
 - Templates and examples
- CircleCI node
 - Operations
 - Templates and examples
- Webex by Cisco node
 - Operations
 - Templates and examples
- Clearbit node
 - Operations
 - Templates and examples
- ClickUp node
 - Operations
 - Operation details
 - Templates and examples
- Clockify node
 - Operations
 - Templates and examples
- Cloudflare node
 - Operations
 - Templates and examples
 - Related resources
- Cockpit node
 - Operations
 - Templates and examples
- Coda node
 - Operations
 - Templates and examples
- CoinGecko node
 - Operations
 - Templates and examples
- Contentful node

- Operations
- Templates and examples
- ConvertKit node
 - Operations
 - Templates and examples
- Copper node
 - Operations
 - Templates and examples
- Cortex node
 - Operations
 - Templates and examples
- CrateDB node
 - Operations
 - Templates and examples
 - Node reference
- crowd.dev node
 - Operations
 - Templates and examples
 - Related resources
- Customer.io node
 - Operations
 - Templates and examples
- DeepL node
 - Operations
 - Templates and examples
- Demio node
 - Operations
 - Templates and examples
- DHL node
 - Operations
 - Templates and examples
- Discord node
 - Operations
 - Templates and examples
 - Related resources
 - Common issues
- Discord node common issues
 - Add extra fields to embeds
 - Mention users and channels
- Discourse node
 - Operations
 - Templates and examples
- Disqus node
 - Operations
 - Templates and examples
- Drift node
 - Operations
 - Templates and examples
- Dropbox node
 - Operations
 - Templates and examples
- Dropcontact node
 - Operations
 - Templates and examples
- E-goi node
 - Operations
 - Templates and examples
- Elasticsearch node
 - Operations
 - Templates and examples

- Elastic Security node
 - Operations
 - Templates and examples
- Emelia node
 - Operations
 - Templates and examples
- ERPNext node
 - Operations
 - Templates and examples
- Facebook Graph API node
 - Operations
 - Templates and examples
- FileMaker node
 - Operations
 - Templates and examples
- Flow node
 - Operations
 - Templates and examples
- Freshdesk node
 - Operations
 - Templates and examples
- Freshservice node
 - Operations
 - Templates and examples
- Freshworks CRM node
 - Operations
 - Templates and examples
- GetResponse node
 - Operations
 - Templates and examples
- Ghost node
 - Operations
 - Templates and examples
- GitHub node
 - Operations
 - Templates and examples
- GitLab node
 - Operations
 - Templates and examples
 - Related resources
- Gmail node
 - Operations
 - Templates and examples
 - Related resources
 - Common issues
- Gmail node Draft Operations
 - Create a draft
 - Delete a draft
 - Get a draft
 - Get Many drafts
 - Common issues
- Gmail node Label Operations
 - Create a label
 - Delete a label
 - Get a label
 - Get Many labels
 - Common issues
- Gmail node Message Operations
 - Add Label to a message
 - Delete a message

- Get a message
- Get Many messages
- Mark as Read
- Mark as Unread
- Remove Label from a message
- Reply to a message
- Send a message
- Send a message and wait for approval
- Common issues
- Gmail node Thread Operations
 - Add Label to a thread
 - Delete a thread
 - Get a thread
 - Get Many threads
 - Remove label from a thread
 - Reply to a message
 - Trash a thread
 - Untrash a thread
 - Common issues
- Gmail node common issues
 - Remove the n8n attribution from sent messages
 - Forbidden - perhaps check your credentials
 - 401 unauthorized error
 - Bad request - please check your parameters
- Gong node
 - Operations
 - Templates and examples
 - Related resources
- Google Ads node
 - Operations
 - Templates and examples
 - Related resources
- Google Analytics node
 - Operations
 - Templates and examples
 - Related resources
- Google BigQuery node
 - Operations
 - Templates and examples
 - Related resources
- Google Books node
 - Operations
 - Templates and examples
- Google Business Profile node
 - Operations
 - Templates and examples
 - Related resources
- Google Calendar node
 - Operations
 - Templates and examples
 - Related resources
- Google Calendar Calendar operations
 - Availability
- Google Calendar Event operations
 - Create
 - Delete
 - Get
 - Get Many
 - Update
- Google Chat node

- Operations
- Templates and examples
- Google Cloud Firestore node
 - Operations
 - Templates and examples
- Google Cloud Natural Language node
 - Operations
 - Templates and examples
- Google Cloud Realtime Database node
 - Operations
 - Templates and examples
- Google Cloud Storage node
 - Operations
 - Templates and examples
 - Related resources
- Google Contacts node
 - Operations
 - Templates and examples
- Google Docs node
 - Operations
 - Templates and examples
- Google Drive node
 - Operations
 - Templates and examples
 - Common issues
- Google Drive File operations
 - Copy a file
 - Create from text
 - Delete a file
 - Download a file
 - Move a file
 - Share a file
 - Update a file
 - Upload a file
- Google Drive File and Folder operations
 - Search files and folders
- Google Drive Folder operations
 - Create a folder
 - Delete a folder
 - Share a folder
- Google Drive Shared Drive operations
 - Create a shared drive
 - Delete a shared drive
 - Get a shared drive
 - Get many shared drives
 - Update a shared drive
- Google Drive node common issues
 - Google hasn't verified this app
 - Google Cloud app becoming unauthorized
 - Google Drive OAuth error
 - Get recent files from Google Drive
- Google Gemini node
 - Operations
 - Templates and examples
 - Related resources
- Google Perspective node
 - Operations
 - Templates and examples
- Google Sheets
 - Operations

- Templates and examples
- Related resources
- Common issues
- Google Sheets Document operations
 - Create a spreadsheet
 - Delete a spreadsheet
- Google Sheets Sheet Within Document operations
 - Append or Update Row
 - Append Row
 - Clear a sheet
 - Create a new sheet
 - Delete a sheet
 - Delete Rows or Columns
 - Get Row(s)
 - Update Row
- Google Sheets node common issues
 - Append an array
 - Column names were updated after the node's setup
- Google Slides node
 - Operations
 - Templates and examples
- Google Tasks node
 - Operations
 - Templates and examples
- Google Translate node
 - Operations
 - Templates and examples
- Google Workspace Admin node
 - Operations
 - Templates and examples
 - How to control which custom fields to fetch for a user
- Gotify node
 - Operations
 - Templates and examples
- GoToWebinar node
 - Operations
 - Templates and examples
- Grafana node
 - Operations
 - Templates and examples
- Grist node
 - Operations
 - Templates and examples
 - Get the Row ID
 - Filter records when using the Get All operation
- Hacker News node
 - Operations
 - Templates and examples
- HaloPSA node
 - Operations
 - Templates and examples
- Harvest node
 - Operations
 - Templates and examples
- Help Scout node
 - Operations
 - Templates and examples
- HighLevel node
 - Operations
 - Templates and examples

- Related resources
- Home Assistant node
 - Operations
 - Templates and examples
 - Related resources
- HubSpot node
 - Operations
 - Templates and examples
- Humantic AI node
 - Operations
 - Templates and examples
- Hunter node
 - Operations
 - Templates and examples
- Intercom node
 - Operations
 - Templates and examples
- Invoice Ninja node
 - Operations
 - Templates and examples
- Iterable node
 - Operations
 - Templates and examples
- Jenkins node
 - Operations
 - Templates and examples
- Jina AI node
 - Operations
 - Templates and examples
 - Related resources
- Jira Software node
 - Operations
 - Templates and examples
 - Related resources
 - Fetch issues for a specific project
- Kafka node
 - Operations
 - Templates and examples
- Keap node
 - Operations
 - Templates and examples
- Kitemaker node
 - Operations
 - Templates and examples
- KoboToolbox node
 - Operations
 - Templates and examples
 - Options
- Lemlist node
 - Operations
 - Templates and examples
- Line node
 - Operations
 - Templates and examples
- Linear node
 - Operations
 - Templates and examples
- LingvaNex node
 - Operations
 - Templates and examples

- LinkedIn node
 - Operations
 - Parameters
 - Templates and examples
 - Related resources
- LoneScale node
 - Operations
 - Templates and examples
 - Related resources
- Magento 2 node
 - Operations
 - Templates and examples
- Mailcheck node
 - Operations
 - Templates and examples
- Mailchimp node
 - Operations
 - Templates and examples
- MailerLite node
 - Operations
 - Templates and examples
- Mailgun node
 - Operations
 - Templates and examples
- Mailjet node
 - Operations
 - Templates and examples
- Mandrill node
 - Operations
 - Templates and examples
- marketstack node
 - Operations
 - Templates and examples
- Matrix node
 - Operations
 - Templates and examples
- Mattermost node
 - Operations
 - Templates and examples
 - Related resources
 - Channel ID field error
 - Find the channel ID
- Mautic node
 - Operations
 - Templates and examples
- Medium node
 - Operations
 - Templates and examples
- MessageBird node
 - Operations
 - Templates and examples
- Metabase node
 - Operations
 - Templates and examples
- Microsoft Dynamics CRM node
 - Operations
 - Templates and examples
- Microsoft Entra ID node
 - Operations
 - Templates and examples

- Related resources
- Common issues
- Microsoft Excel 365 node
 - Operations
 - Templates and examples
- Microsoft Graph Security node
 - Operations
 - Templates and examples
- Microsoft OneDrive node
 - Operations
 - Templates and examples
 - Related resources
 - Find the folder ID
- Microsoft Outlook node
 - Operations
 - Templates and examples
 - Related resources
- Microsoft SharePoint node
 - Operations
 - Templates and examples
 - Related resources
- Microsoft SQL node
 - Operations
 - Templates and examples
- Microsoft Teams node
 - Operations
 - Templates and examples
 - Related resources
- Microsoft To Do node
 - Operations
 - Templates and examples
- Mindee node
 - Operations
 - Templates and examples
- MISP node
 - Operations
 - Templates and examples
- Mistral AI node
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- Mocean node
 - Operations
 - Templates and examples
- monday.com node
 - Operations
 - Templates and examples
- MongoDB node
 - Operations
 - Templates and examples
- Monica CRM node
 - Operations
 - Templates and examples
- MQTT node
 - Operations
 - Templates and examples
 - Related resources
- MSG91 node
 - Operations

- Templates and examples
- Find your Sender ID
- MySQL node
 - Operations
 - Templates and examples
 - Related resources
 - Use query parameters
 - Common issues
- MySQL node common issues
 - Update rows by composite key
 - Can't connect to a local MySQL server when using Docker
 - Decimal numbers returned as strings
- Customer Datastore (n8n Training) node
- Customer Messenger (n8n Training) node
- NASA node
 - Operations
 - Templates and examples
- Netlify node
 - Operations
 - Templates and examples
- Netscaler ADC node
 - Operations
 - Templates and examples
 - Related resources
- Nextcloud node
 - Operations
 - Templates and examples
- NocoDB node
 - Operations
 - Templates and examples
 - Relates resources
- Notion node
 - Operations
 - Templates and examples
 - Related resources
 - Common issues
- Notion node common issues
 - Relation property not displaying
 - Create toggle heading
 - Handle null and empty values
- npm node
 - Operations
 - Templates and examples
 - Related resources
- Odoo node
 - Operations
 - Templates and examples
- Okta node
 - Operations
 - Templates and examples
 - Related resources
- One Simple API node
 - Operations
 - Templates and examples
 - Related resources
- Onfleet node
 - Operations
 - Templates and examples
- OpenAI node
 - Operations

- Templates and examples
- Related resources
- Using tools with OpenAI assistants
- Common issues
- OpenAI Assistant operations
 - Create an Assistant
 - Delete an Assistant
 - List Assistants
 - Message an Assistant
 - Update an Assistant
 - Common issues
- OpenAI Audio operations
 - Generate Audio
 - Transcribe a Recording
 - Translate a Recording
 - Common issues
- OpenAI Conversation operations
 - Create a Conversation
 - Get a Conversation
 - Remove a Conversation
 - Update a Conversation
- OpenAI File operations
 - Delete a File
 - List Files
 - Upload a File
 - Common issues
- OpenAI Image operations
 - Analyze Image
 - Generate an Image
 - Edit an Image
 - Common issues
- OpenAI Text operations
 - Generate a Chat Completion
 - Generate a Model Response
 - Classify Text for Violations
 - Common issues
- OpenAI Video operations
 - Generate Video
- OpenAI node common issues
- OpenThesaurus node
 - Operations
 - Templates and examples
- OpenWeatherMap node
 - Operations
 - Templates and examples
- Oracle Database node
 - Operations
 - Related resources
 - Use bind parameters
 - Use n8n Expressions for bind values
 - VARCHAR
- Oura node
 - Operations
 - Templates and examples
- Paddle node
 - Operations
 - Templates and examples
- PagerDuty node
 - Operations
 - Templates and examples

- PayPal node
 - Operations
 - Templates and examples
- Peekalink node
 - Operations
 - Templates and examples
- Perplexity node
 - Operations
 - Templates and examples
 - Related resources
- PhantomBuster node
 - Operations
 - Templates and examples
- Philips Hue node
 - Operations
 - Templates and examples
- Pipedrive node
 - Operations
 - Templates and examples
- Plivo node
 - Operations
 - Templates and examples
- PostBin node
 - Operations
 - Templates and examples
 - Send requests
 - Create and manage bins
- Postgres node
 - Operations
 - Templates and examples
 - Related resources
 - Use query parameters
 - Common issues
- Postgres node common issues
 - Dynamically populate SQL IN groups with parameters
 - Working with timestamps and time zones
 - Outputting Date columns as date strings instead of ISO datetime strings
- PostHog node
 - Operations
 - Templates and examples
- ProfitWell node
 - Operations
 - Templates and examples
- Pushbullet node
 - Operations
 - Templates and examples
- Pushcut node
 - Operations
 - Templates and examples
- Pushover node
 - Operations
 - Templates and examples
- QuestDB node
 - Operations
 - Templates and examples
 - Node reference
- Quick Base node
 - Operations
 - Templates and examples

- QuickBooks Online node
 - Operations
 - Templates and examples
- QuickChart node
 - Operations
 - Templates and examples
 - Related resources
- RabbitMQ node
 - Operations
 - Templates and examples
- Raindrop node
 - Operations
 - Templates and examples
- Reddit node
 - Operations
 - Templates and examples
- Redis node
 - Operations
 - Templates and examples
- Rocket.Chat node
 - Operations
 - Templates and examples
- Rundeck node
 - Operations
 - Templates and examples
 - Find the job ID
- S3 node
 - Operations
 - Templates and examples
 - Node reference
- Salesforce node
 - Operations
 - Templates and examples
 - Working with Salesforce custom fields
- Salesmate node
 - Operations
 - Templates and examples
- SeaTable node
 - Operations
 - Templates and examples
- SecurityScorecard node
 - Operations
 - Templates and examples
- Segment node
 - Operations
 - Templates and examples
- SendGrid node
 - Operations
 - Templates and examples
- Sendy node
 - Operations
 - Templates and examples
- Sentry.io node
 - Operations
 - Templates and examples
 - Related resources
- ServiceNow node
 - Operations
 - Templates and examples
- seven node

- Operations
- Templates and examples
- Shopify node
 - Operations
 - Templates and examples
- SIGNL4 node
 - Operations
 - Templates and examples
- Slack node
 - Operations
 - Templates and examples
 - Related resources
 - Required scopes
- Snowflake node
 - Operations
 - Templates and examples
- Splunk node
 - Operations
 - Templates and examples
- Spotify node
 - Operations
 - Templates and examples
- Stackby node
 - Operations
 - Templates and examples
- Storyblok node
 - Operations
 - Templates and examples
- Strapi node
 - Operations
 - Templates and examples
 - Related resources
- Strava node
 - Operations
 - Templates and examples
- Stripe node
 - Operations
 - Templates and examples
- Supabase node
 - Operations
 - Using custom schemas
 - Templates and examples
 - Common issues
- Supabase node common issues
 - Filtering rows by metadata
 - Can't connect to a local Supabase database when using Docker
 - Records are accessible through Postgres but not Supabase
- SyncroMSP node
 - Operations
 - Templates and examples
- Taiga node
 - Operations
 - Templates and examples
- Tapfiliate node
 - Operations
 - Templates and examples
- Telegram node
 - Operations
 - Related resources
 - Common issues

- Telegram node Chat operations
 - Get Chat
 - Get Administrators
 - Get Chat Member
 - Leave Chat
 - Set Description
 - Set Title
- Telegram node Callback operations
 - Answer Query
 - Answer Inline Query
- Telegram node File operations
 - Get File
- Telegram node Message operations
 - Edit Message Text
 - Pin Chat Message
 - Send Animation
 - Send Chat Action
 - Send Document
 - Send Location
 - Send Media Group
 - Send Message
 - Send and Wait for Response
 - Send Photo
 - Send Sticker
 - Send Video
 - Unpin Chat Message
 - Reply Markup parameters
- Telegram node common issues
 - Add a bot to a Telegram channel
 - Get the Chat ID
 - Send more than 30 messages per second
 - Remove the n8n attribution from sent messages
- TheHive node
 - Operations
 - Templates and examples
 - Related resources
- TheHive 5 node
 - Operations
 - Templates and examples
 - Related resources
- TimescaleDB node
 - Operations
 - Templates and examples
 - Specify a column's data type
- Todoist node
 - Operations
 - Templates and examples
- Travis CI node
 - Operations
 - Templates and examples
- Trello node
 - Operations
 - Templates and examples
 - Find the List ID
- Twake node
 - Operations
 - Templates and examples
- Twilio node
 - Operations
 - Templates and examples

Related resources

Twist node

- Operations
- Templates and examples
- Get the User ID

Unleashed Software node

- Operations
- Templates and examples

UpLead node

- Operations
- Templates and examples

uProc node

- Operations
- Templates and examples

UptimeRobot node

- Operations
- Templates and examples

urlscan.io node

- Operations
- Templates and examples

Venafi TLS Protect Cloud node

- Operations
- Templates and examples
- Related resources

Venafi TLS Protect Datacenter node

- Operations
- Templates and examples
- Related resources

Vero node

- Operations
- Templates and examples

Vonage node

- Operations
- Templates and examples

Webflow node

- Operations
- Templates and examples

Wekan node

- Operations
- Templates and examples
- Load all the parameters for the node

WhatsApp Business Cloud node

- Operations
- Templates and examples
- Related resources
- Common issues

WhatsApp Business Cloud node common issues

- Bad request - please check your parameters
- Working with non-text media

Wise node

- Operations
- Templates and examples

WooCommerce node

- Operations
- Templates and examples

WordPress node

- Operations
- Templates and examples

X (Formerly Twitter) node

- Operations

- Templates and examples
- Xero node
 - Operations
 - Templates and examples
 - Related resources
- Yourls node
 - Operations
 - Templates and examples
- YouTube node
 - Operations
 - Templates and examples
- Zammad node
 - Operations
 - Templates and examples
- Zendesk node
 - Operations
 - Templates and examples
- Zoho CRM node
 - Operations
 - Templates and examples
- Zoom node
 - Operations
 - Templates and examples
- Zulip node
 - Operations
 - Templates and examples
- Triggers library
- ActiveCampaign Trigger node
 - Events
 - Related resources
- Acuity Scheduling Trigger node
 - Events
- Affinity Trigger node
 - Events
 - Related resources
- Airtable Trigger node
 - Events
 - Related resources
 - Node parameters
- AMQP Trigger node
- Asana Trigger node
 - Events
 - Related resources
- Autopilot Trigger node
 - Events
- AWS SNS Trigger node
 - Events
 - Related resources
- Bitbucket Trigger node
- Box Trigger node
 - Find your Box Target ID
- Brevo Trigger node
 - Events
 - Related resources
- Calendly Trigger node
 - Events
- Cal Trigger node
 - Events
- Chargebee Trigger node
 - Add webhook URL in Chargebee

- ClickUp Trigger node
 - Events
 - Related resources
- Clockify Trigger node
- ConvertKit Trigger node
 - Events
 - Related resources
- Copper Trigger node
 - Events
 - Related resources
- crowd.dev Trigger node
 - Events
 - Related resources
- Customer.io Trigger node
 - Events
 - Related resources
- Emelia Trigger node
 - Events
- Eventbrite Trigger node
- Facebook Lead Ads Trigger node
 - Events
 - Related resources
 - Common issues
- Facebook Trigger node
 - Objects
 - Related resources
- Facebook Trigger Ad Account object
 - Trigger configuration
 - Related resources
- Facebook Trigger Application object
 - Trigger configuration
 - Related resources
- Facebook Trigger Certificate Transparency object
 - Trigger configuration
 - Related resources
- Facebook Trigger Group object
 - Trigger configuration
 - Related resources
- Facebook Trigger Instagram object
 - Trigger configuration
 - Related resources
- Facebook Trigger Link object
 - Trigger configuration
 - Related resources
- Facebook Trigger Page object
 - Prerequisites
 - Trigger configuration
 - Related resources
- Facebook Trigger Permissions object
 - Trigger configuration
 - Related resources
- Facebook Trigger User object
 - Trigger configuration
 - Related resources
- Facebook Trigger WhatsApp Business Account object
 - Prerequisites
 - Trigger configuration
- Facebook Trigger Workplace Security object
 - Trigger configuration
 - Related resources

- Figma Trigger (Beta) node
 - Events
- Flow Trigger node
 - Events
 - Related resources
- Form.io Trigger node
- Formstack Trigger node
- GetResponse Trigger node
 - Events
- GitHub Trigger node
 - Events
 - Related resources
- GitLab Trigger node
 - Events
 - Related resources
- Gmail Trigger node
 - Events
 - Node parameters
 - Node filters
 - Related resources
 - Common issues
- Gmail Trigger node Poll Mode options
 - Poll mode options
- Gmail Trigger node common issues
 - 401 unauthorized error
- Google Calendar Trigger node
 - Events
 - Related resources
- Google Drive Trigger node
 - Common issues
- Google Drive Trigger node common issues
 - 401 unauthorized error
 - Handling more than one file change
- Google Business Profile Trigger node
 - Events
 - Related resources
- Google Sheets Trigger node
 - Events
 - Related resources
 - Common issues
- Google Sheets Trigger node common issues
 - Stuck waiting for trigger event
 - Date and time columns are rendering as numbers
- Gumroad Trigger node
- Help Scout Trigger node
- HubSpot Trigger node
 - Events
 - Related resources
- Invoice Ninja Trigger node
- Jira Trigger node
- JotForm Trigger node
- Kafka Trigger node
- Keap Trigger node
- KoboToolbox Trigger node
- Lemlist Trigger node
 - Events
- Linear Trigger node
 - Events
- LoneScale Trigger node
 - Events

- Related resources
- Mailchimp Trigger node
- MailerLite Trigger node
 - Events
- Mailjet Trigger node
- Mautic Trigger node
 - Related resources
- Microsoft OneDrive Trigger node
 - Events
- Related resources
- Microsoft Outlook Trigger node
 - Events
- Related resources
- Microsoft Teams Trigger node
 - Events
- Related resources
- MQTT Trigger node
- Netlify Trigger node
 - Related resources
- Notion Trigger node
 - Events
- Related resources
- Onfleet Trigger node
 - Events
- PayPal Trigger node
- Pipedrive Trigger node
- Postgres Trigger node
 - Events
- Related resources
- Postmark Trigger node
- Pushcut Trigger node
 - Configure a Pushcut action
- RabbitMQ Trigger node
 - Related resources
- Redis Trigger node
- Salesforce Trigger node
 - Events
- Related resources
- SeaTable Trigger node
- Shopify Trigger node
- Slack Trigger node
 - Events
- Parameters
- Options
- Related resources
- Required scopes
- Verify the webhook
- Common issues
- Strava Trigger node
 - Events
- Stripe Trigger node
- SurveyMonkey Trigger node
- Taiga Trigger node
- Telegram Trigger node
 - Events
- Options
- Related resources
- Common issues
- Telegram Trigger node common issues
 - Stuck waiting for trigger event

Bad request: bad webhook: An HTTPS URL must be provided for webhook
Workflow only works in testing or production

TheHive 5 Trigger node

- Events
- Related resources
- Configure a webhook in TheHive

TheHive Trigger node

- Events
- Related resources
- Configure a webhook in TheHive

Toggl Trigger node

Trello Trigger node

- Find the Model ID

Twilio Trigger node

- Events
- Related resources

Typeform Trigger node

Venafi TLS Protect Cloud Trigger node

Webex by Cisco Trigger node

Webflow Trigger node

WhatsApp Trigger node

- Events
- Related resources
- Common issues

Wise Trigger node

- Events

WooCommerce Trigger node

- Events

Workable Trigger node

- Events
- Related resources

Wufoo Trigger node

Zendesk Trigger node

Cluster nodes

- Root nodes
- Sub-nodes

Root nodes

AI Agent node

- Templates and examples
- Related resources
- Common issues

Conversational AI Agent node

- Node parameters
- Node options
- Templates and examples
- Common issues

OpenAI Functions Agent node

- Node parameters
- Node options
- Templates and examples
- Common issues

Plan and Execute Agent node

- Node parameters
- Node options
- Templates and examples
- Common issues

ReAct AI Agent node

- Node parameters
- Node options

- Related resources
- Templates and examples
- Common issues
- SQL AI Agent node
 - Node parameters
 - Node options
 - Templates and examples
 - Common issues
- Tools AI Agent node
 - Node parameters
 - Node options
 - Templates and examples
 - Dynamic parameters for tools with `$fromAI()`
 - Common issues
- AI Agent node common issues
 - Internal error: 400 Invalid value for 'content'
 - Error in sub-node Simple Memory
 - A Chat Model sub-node must be connected error
 - No prompt specified error
- Basic LLM Chain node
 - Node parameters
 - Chat Messages
 - Templates and examples
 - Related resources
 - Common issues
- Question and Answer Chain node
 - Node parameters
 - Templates and examples
 - Related resources
 - Common issues
- Question and Answer Chain node common issues
 - No prompt specified error
 - A Retriever sub-node must be connected error
 - Can't produce longer responses
- Summarization Chain node
 - Node parameters
 - Node Options
 - Templates and examples
 - Related resources
- Information Extractor node
 - Node parameters
 - Node options
 - Related resources
- Text Classifier node
 - Node parameters
 - Node options
 - Related resources
- Sentiment Analysis node
 - Node parameters
 - Node options
 - Usage Notes
 - Example Usage
 - Related resources
- LangChain Code node
 - Node parameters
 - Node inputs and outputs configuration
 - Built-in methods
 - Templates and examples
 - Related resources
- Azure AI Search Vector Store node

- Prerequisites
- Node usage patterns
- Node parameters
- Node options
- Azure AI Search specific features
- Troubleshooting
- Templates and examples
- Related resources
- Simple Vector Store node
 - Data safety limitations
 - Node usage patterns
 - Memory Management
 - Node parameters
 - Templates and examples
 - Related resources
- Milvus Vector Store node
 - Node usage patterns
 - Node parameters
 - Node options
 - Related resources
- MongoDB Atlas Vector Store node
 - Prerequisites
 - Node usage patterns
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- PGVector Vector Store node
 - Node usage patterns
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- Pinecone Vector Store node
 - Node usage patterns
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- Qdrant Vector Store node
 - Node usage patterns
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- Redis Vector Store node
 - Prerequisites
 - Node usage patterns
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- Supabase Vector Store node
 - Node usage patterns
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- Weaviate Vector Store node
 - Node usage patterns

- Node parameters
- Node options
- Templates and examples
- Related resources
- Zep Vector Store node
 - Node usage patterns
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- Sub nodes
- Default Data Loader node
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- GitHub Document Loader node
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- Embeddings AWS Bedrock node
 - Node parameters
 - Templates and examples
 - Related resources
- Embeddings Azure OpenAI node
 - Node options
 - Templates and examples
 - Related resources
- Embeddings Cohere node
 - Node parameters
 - Templates and examples
 - Related resources
- Embeddings Google Gemini node
 - Node parameters
 - Templates and examples
 - Related resources
- Embeddings Google PaLM node
 - Node parameters
 - Templates and examples
 - Related resources
- Embeddings Google Vertex node
 - Node parameters
 - Templates and examples
 - Related resources
- Embeddings HuggingFace Inference node
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- Embeddings Mistral Cloud node
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- Embeddings Ollama node
 - Node parameters
 - Templates and examples
 - Related resources
- Embeddings OpenAI node

- Node options
- Templates and examples
- Related resources
- Anthropic Chat Model node
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- AWS Bedrock Chat Model node
 - Node parameters
 - Node options
 - Proxy limitations
 - Templates and examples
 - Related resources
- Azure OpenAI Chat Model node
 - Node parameters
 - Node options
 - Proxy limitations
 - Templates and examples
 - Related resources
- Cohere Chat Model node
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- DeepSeek Chat Model node
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- Google Gemini Chat Model node
 - Node parameters
 - Node options
 - Limitations
 - Templates and examples
 - Related resources
- Google Vertex Chat Model node
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- Groq Chat Model node
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- Mistral Cloud Chat Model node
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
- Ollama Chat Model node
 - Node parameters
 - Node options
 - Templates and examples
 - Related resources
 - Common issues
- Ollama Chat Model node common issues
 - Processing parameters
 - Can't connect to a remote Ollama instance

Can't connect to a local Ollama instance when using Docker
Error: connect ECONNREFUSED ::1:11434
Ollama and HTTP/HTTPS proxies

OpenAI Chat Model node

- Node parameters
- Node options
- Additional node options (Responses API only)
- Templates and examples
- Related resources
- Common issues

OpenAI Chat Model node common issues

- Processing parameters

OpenRouter Chat Model node

- Node parameters
- Node options
- Templates and examples
- Related resources

Vercel AI Gateway Chat Model node

- Node parameters
- Node options
- Templates and examples
- Related resources

xAI Grok Chat Model node

- Node parameters
- Node options
- Templates and examples
- Related resources

Cohere Model node

- Node Options
- Templates and examples
- Related resources

Ollama Model node

- Node parameters
- Node options
- Templates and examples
- Related resources
- Common issues

Ollama Model node common issues

- Processing parameters
- Can't connect to a remote Ollama instance
- Can't connect to a local Ollama instance when using Docker
- Error: connect ECONNREFUSED ::1:11434

Ollama and HTTP/HTTPS proxies

Hugging Face Inference Model node

- Node parameters
- Node options
- Templates and examples
- Related resources

Chat Memory Manager node

- Node parameters
- Templates and examples
- Related resources

Simple Memory node

- Node parameters
- Templates and examples
- Related resources
- Common issues

Simple Memory node common issues

- Single memory instance
- Managing the Session ID

- Motorhead node**
 - Node parameters
 - Node reference
 - Templates and examples
 - Related resources
 - Single memory instance
- MongoDB Chat Memory node**
 - Node parameters
 - Related resources
 - Single memory instance
- Redis Chat Memory node**
 - Node parameters
 - Templates and examples
 - Related resources
 - Single memory instance
- Postgres Chat Memory node**
 - Node parameters
 - Related resources
 - Single memory instance
- Xata node**
 - Node parameters
 - Templates and examples
 - Related resources
 - Single memory instance
- Zep node**
 - Node parameters
 - Templates and examples
 - Related resources
 - Single memory instance
- Auto-fixing Output Parser node**
 - Templates and examples
 - Related resources
- Item List Output Parser node**
 - Node options
 - Templates and examples
 - Related resources
- Structured Output Parser node**
 - Node parameters
 - Templates and examples
 - Related resources
 - Common issues
- Structured Output Parser node common issues**
 - Processing parameters
 - Adding the structured output parser node to AI nodes
 - Using the structured output parser to format intermediary steps
 - Structuring output from agents
- Contextual Compression Retriever node**
 - Templates and examples
 - Related resources
- MultiQuery Retriever node**
 - Node options
 - Templates and examples
 - Related resources
- Vector Store Retriever node**
 - Node parameters
 - Templates and examples
 - Related resources
- Workflow Retriever node**
 - Node parameters
 - Templates and examples

- Related resources
- Character Text Splitter node
 - Node parameters
 - Templates and examples
 - Related resources
- Recursive Character Text Splitter node
 - Node parameters
 - Templates and examples
 - Related resources
- Token Splitter node
 - Node parameters
 - Templates and examples
 - Related resources
- AI Agent Tool node
 - Node parameters
 - Node options
 - Templates and examples
 - Dynamic parameters for tools with `$fromAI()`
- Calculator node
 - Templates and examples
 - Related resources
- Custom Code Tool node
 - Node parameters
 - Templates and examples
 - Related resources
- MCP Client Tool node
 - Node parameters
 - Templates and examples
 - Related resources
- SearXNG Tool node
 - Node Options
 - Running a SearXNG instance
 - Templates and examples
 - Related resources
- SerpApi (Google Search) node
 - Node options
 - Templates and examples
 - Related resources
- Think Tool node
 - Templates and examples
 - Related resources
- Vector Store Question Answer Tool node
 - Node parameters
 - How n8n populates the tool description
 - Related resources
- Wikipedia node
 - Templates and examples
 - Related resources
- Wolfram|Alpha tool node
 - Templates and examples
 - Related resources
- Call n8n Workflow Tool node
 - Node parameters
 - Templates and examples
 - Related resources
- Reranker Cohere
 - Node parameters
 - Templates and examples
 - Related resources
- Model Selector

- Node parameters
- Templates and examples
- Related resources
- Credentials library
- Action Network credentials
 - Supported authentication methods
 - Related resources
 - Using API key
 - Request API access
- ActiveCampaign credentials
 - Supported authentication methods
 - Related resources
 - Using API key
- Acuity Scheduling credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
 - Using OAuth2
- Adalo credentials
 - Supported authentication methods
 - Related resources
 - Using API key
- Affinity credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Agile CRM credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Airtable credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using personal access token
 - Using OAuth2
- Airtop credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- AlienVault credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- AMQP credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using AMQP connection
- Anthropic credentials
 - Supported authentication methods
 - Related resources
 - Using API key
- APITemplate.io credentials
 - Prerequisites

- Supported authentication methods
- Related resources
- Using API key
- Asana credentials
 - Supported authentication methods
 - Related resources
 - Using Access token
 - Using OAuth2
- Auth0 Management credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API client secret
- Autopilot credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- AWS credentials
 - AWS (IAM) credentials
 - AWS (Assume Role) credentials
- Azure OpenAI credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
 - Using Azure Entra ID (OAuth2)
 - Setting custom scopes
- Azure Cosmos DB credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API Key
 - Common issues
- Azure AI Search credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
 - Troubleshooting
- Azure Storage credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using OAuth2
 - Using Shared Key
 - Common issues
- BambooHR credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API Key
- Bannerbear credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Baserow credentials
 - Prerequisites
 - Supported authentication methods

- Related resources
- Using basic auth
- Beeminder credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API user token
- Bitbucket credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Configuring Bitbucket access token
- Bitly credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API token
 - Using OAuth2
- Bitwarden credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Box credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using OAuth2
- Brandfetch credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Brevo credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - API key
- Bubble credentials
 - Supported authentication methods
 - Related resources
 - Using API key
- Cal.com credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Calendly credentials
 - Supported authentication methods
 - Related resources
 - Using API access token
 - Using OAuth2
- Carbon Black credentials
 - Prerequisites
 - Authentication methods
 - Related resources
 - Using API key
- Chargebee credentials
 - Prerequisites
 - Supported authentication methods

- Related resources
- Using API key
- CircleCI credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using personal API token
- Cisco Meraki credentials
 - Prerequisites
 - Authentication methods
 - Related resources
 - Using API key
- Cisco Secure Endpoint credentials
 - Prerequisites
 - Authentication methods
 - Related resources
 - Using OAuth2
- Cisco Umbrella credentials
 - Prerequisites
 - Authentication methods
 - Related resources
 - Using API key
- Clearbit credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- ClickUp credentials
 - Supported authentication methods
 - Related resources
 - Using API access token
 - Using OAuth2
- Clockify credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Cloudflare credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API token
- Cockpit credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API access token
- Coda credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API access token
- Cohere credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Contentful credentials
 - Prerequisites
 - Supported authentication methods

- Related resources
- Using API access token
- ConvertAPI credentials
 - Supported authentication methods
 - Related resources
 - Using API Token
- ConvertKit credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Copper credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Cortex credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- CrateDB credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using account connection
- crowd.dev credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- CrowdStrike credentials
 - Prerequisites
 - Authentication methods
 - Related resources
 - Using OAuth2
- Customer.io credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
 - Why you need a Tracking API Key and an App API Key
- Datadog credentials
 - Prerequisites
 - Related resources
 - Using API Key
- DeepL credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- DeepSeek credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Demio credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources

- Using API key
- DFIR-IRIS credentials
 - Prerequisites
 - Related resources
 - Using API Key
- DHL credentials
 - Supported authentication methods
 - Related resources
 - Using API key
- Discord credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using bot
 - Using OAuth2
 - Using webhook
 - Choose an authentication method
- Discourse credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Disqus credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API access token
- Drift credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API personal access token
 - Using OAuth2
- Dropbox credentials
 - Supported authentication methods
 - Related resources
 - Using access token
 - Using OAuth2
- Dropcontact credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Dynatrace credentials
 - Prerequisites
 - Related resources
 - Using Access Token
- E-goi credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Elasticsearch credentials
 - Supported authentication methods
 - Related resources
 - Using basic auth
- Elastic Security credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources

- Using basic auth
- Using API key

Emelia credentials

- Prerequisites
- Supported authentication methods
- Related resources
- Using API key

ERPNext credentials

- Prerequisites
- Supported authentication methods
- Related resources
- Using API key
- How to find the subdomain of an ERPNext cloud-hosted account

Eventbrite credentials

- Prerequisites
- Supported authentication methods
- Related resources
- Using API private key
- Using OAuth2

F5 Big-IP credentials

- Prerequisites
- Authentication methods
- Related resources
- Using account login

Facebook App credentials

- Supported authentication methods
- Related resources
- Using app access token
- Common issues

Facebook Graph API credentials

- Supported authentication methods
- Related resources
- Using app access token

Facebook Lead Ads credentials

- Supported authentication methods
- Related resources
- Using OAuth2

Figma credentials

- Prerequisites
- Supported authentication methods
- Related resources
- Using API key

FileMaker credentials

- Prerequisites
- Supported authentication methods
- Related resources
- Using database connection

Filescan credentials

- Prerequisites
- Related resources
- Using API key

Flow credentials

- Prerequisites
- Supported authentication methods
- Related resources
- Using API key

Form.io Trigger credentials

- Supported authentication methods
- Related resources
- Using basic auth

Formstack Trigger credentials
 Prerequisites
 Supported authentication methods
 Related resources
 Using API access token
 Using OAuth2

Fortinet FortiGate credentials
 Prerequisites
 Supported authentication methods
 Related resources
 Using API access token

Freshdesk credentials
 Prerequisites
 Supported authentication methods
 Related resources
 Using API key

Freshservice credentials
 Prerequisites
 Supported authentication methods
 Related resources
 Using API key

Freshworks CRM credentials
 Prerequisites
 Supported authentication methods
 Related resources
 Using API key

FTP credentials
 Prerequisites
 Supported authentication methods
 Related resources
 Using FTP account
 Using SFTP account

GetResponse credentials
 Prerequisites
 Supported authentication methods
 Related resources
 Using API key
 Using OAuth2
 Configure OAuth2 credentials for a local environment

Ghost credentials
 Prerequisites
 Supported authentication methods
 Related resources
 Using Admin API key
 Using Content API key

Git credentials
 Prerequisites
 Supported authentication methods
 Related resources
 Using basic auth

GitHub credentials
 Prerequisites
 Supported authentication methods
 Related resources
 Using API access token
 Using OAuth2

GitLab credentials
 Supported authentication methods
 Related resources
 Using API access token

Using OAuth2
Gong credentials
 Supported authentication methods
 Related resources
 Using API access token
 Using OAuth2
Google credentials
 OAuth2 and Service Account
 Compatible nodes
Google: OAuth2 single service
 Prerequisites
 Set up OAuth
 Video
 Troubleshooting
Google: OAuth2 generic
 Prerequisites
 Set up OAuth
 Video
 Scopes
 Troubleshooting
Google: Service Account
 Prerequisites
 Set up Service Account
 Video
 Troubleshooting
Google Gemini(PaLM) credentials
 Prerequisites
 Supported authentication methods
 Related resources
 Using Gemini(PaLM) API key
Gotify credentials
 Prerequisites
 Supported authentication methods
 Related resources
 Using API token
GoTo Webinar credentials
 Prerequisites
 Supported authentication methods
 Related resources
 Using OAuth2
Grafana credentials
 Prerequisites
 Supported authentication methods
 Related resources
 Using API key
Grist credentials
 Prerequisites
 Supported authentication methods
 Related resources
 Using API key
Groq credentials
 Prerequisites
 Supported authentication methods
 Related resources
 Using API key
Gumroad credentials
 Prerequisites
 Supported authentication methods
 Related resources
 Using API access token

- HaloPSA credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Harvest credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API Access Token
 - Using OAuth2
- Help Scout credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using OAuth2
- HighLevel credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
 - Using OAuth2
- Home Assistant credentials
 - Supported authentication methods
 - Related resources
 - Using API access token
- HTTP Request credentials
 - Prerequisites
 - Supported authentication methods
 - Using predefined credential type
 - Using OAuth1
 - Using OAuth2
 - Using query auth
 - Using custom auth
 - Provide an SSL certificate
- HubSpot credentials
 - Supported authentication methods
 - Related resources
 - Using App token
 - Using Developer API key
 - Using OAuth2
 - Required scopes for HubSpot node
- Hugging Face credentials
 - Supported authentication methods
 - Related resources
 - Using API key
- Humanotic AI credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Hunter credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Hybrid Analysis credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources

- Using API key
- IMAP credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using user account
- Gmail IMAP credentials
 - Prerequisites
 - Set up the credential
- Outlook.com IMAP credentials
 - Set up the credentials
 - Connection errors
 - Use an app password
- Yahoo IMAP credentials
 - Prerequisites
 - Set up the credential
- Imperva WAF credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Intercom credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Invoice Ninja credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Iterable credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Jenkins credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API token
- Jina AI credentials
 - Supported authentication methods
 - Related resources
 - Using API key
- Jira credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using SW Cloud API token
 - Using SW Server account
- JotForm credentials
 - Supported authentication methods
 - Related resources
 - Using API key
- JWT credentials
 - Supported authentication methods
 - Related resources
 - Using Passphrase
 - Using private key (PEM key)

- Available algorithms
- Kafka credentials
 - Supported authentication methods
 - Related resources
 - Using client ID
- Keap credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using OAuth2
- Kibana credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using basic auth
- Kitemaker credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API access token
- KoboToolbox credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API token
- LDAP credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using LDAP server details
- Lemlist credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Line credentials
 - Supported authentication methods
 - Related resources
 - Using Notify OAuth2
- Linear credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
 - Using OAuth2
- LingvaNex credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- LinkedIn credentials
 - Prerequisites
 - Supported authentication methods
 - Related Resources
 - Using Community Management OAuth2
 - Using OAuth2
- LoneScale credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources

- Using API key
- Magento 2 credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API access token
- Mailcheck credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API Key
- Mailchimp credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
 - Using OAuth2
 - Selecting an authentication method
- MailerLite credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Mailgun credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
 - Working with multiple email domains
- Mailjet credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using Email API key
 - Using SMS Token
- Malcore credentials
 - Prerequisites
 - Related resources
 - Using API key
- Mandrill credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Marketstack credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Matrix credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API access token
- Mattermost credentials
 - Supported authentication methods
 - Related resources
 - Using API access token
 - Enable personal access tokens
- Mautic credentials

- Supported authentication methods
- Related resources
- Using basic auth
- Using OAuth2
- Enable the API
- Medium credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API access token
 - Using OAuth2
- MessageBird credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Metabase credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using basic auth
- Microsoft credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using OAuth2
 - Common issues
- Microsoft Azure Monitor credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using OAuth2
- Microsoft Entra ID credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using OAuth2
 - Setting custom scopes
 - Common issues
- Microsoft SQL credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using SQL database connection
- Milvus credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using basic auth
- Mindee credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using invoice API key
 - Using receipt API key
- Miro credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using OAuth2

- MISP credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Mist credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API token
- Mistral Cloud credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Mocean credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- monday.com credentials
 - Supported authentication methods
 - Related resources
 - Using API token
 - Using OAuth2
- MongoDB credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using database connection - Connection string
 - Using database connection - Values
- Monica CRM credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API token
- Motorhead credentials
 - Supported authentication methods
 - Related resources
 - Using API key
- MQTT credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using broker connection
- MSG91 credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
 - IP Security
- MySQL credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using database connection
- NASA credentials
 - Supported authentication methods
 - Related resources
 - Using an API key

- Netlify credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API access token
- Netscaler ADC credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using basic auth
- Nextcloud credentials
 - Supported authentication methods
 - Related resources
 - Using basic auth
 - Using OAuth2
- NocoDB credentials
 - Supported authentication methods
 - Related resources
 - Using API token
 - Using user auth token
- Notion credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API integration token
 - Using OAuth2
 - Internal vs. public integrations
- npm credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API access token
- Odoo credentials
 - Supported authentication methods
 - Related resources
 - Using API key
 - Using password
 - Required plan type
- Okta credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using SSWS API access token
- Ollama credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using instance URL
- One Simple API credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API token
- Onfleet credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- OpenAI credentials
 - Prerequisites

- Supported authentication methods
- Related resources
- Using API key
- OpenCTI credentials
 - Prerequisites
 - Authentication methods
 - Related resources
 - Using API key
- OpenRouter credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- OpenWeatherMap credentials
 - Supported authentication methods
 - Related resources
 - Using API access token
- Oracle Database credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using database connection
- Oura credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API access token
- Paddle credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API access token (Classic)
- PagerDuty credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API token
 - Using OAuth2
- PayPal credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API client and secret
- Peekalink credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Perplexity credentials
 - Supported authentication methods
 - Related resources
 - Using API key
- PhantomBuster credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Philips Hue credentials
 - Prerequisites
 - Supported authentication methods

- Related resources
- Using OAuth2
- Pinecone credentials
 - Supported authentication methods
 - Related resources
 - Using API key
- Pipedrive credentials
 - Supported authentication methods
 - Related resources
 - Using API token
 - Using OAuth2
- Plivo credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using basic auth
- Postgres credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using database connection
- PostHog credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Postmark credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API token
- ProfitWell credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API token
- Pushbullet credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using OAuth2
- Pushcut credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Pushover credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API Key
- QRadar credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Qdrant credentials
 - Supported authentication methods
 - Related resources
 - Using API key

- Qualys credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using basic auth
- QuestDB credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using database connection
- Quick Base credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- QuickBooks credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using OAuth2
- RabbitMQ credentials
 - Supported authentication methods
 - Related resources
 - Using user connection
 - guest user issues
- Raindrop credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using OAuth
- Rapid7 InsightVM credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Recorded Future credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API access token
- Reddit credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using OAuth2
- Redis credentials
 - Supported authentication methods
 - Related resources
 - Using database connection
- Rocket.Chat credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API access token
- Rundeck credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API token
- S3 credentials

- Prerequisites
- Supported authentication methods
- Related resources
- Using S3 endpoint
- Salesforce credentials
 - Supported authentication methods
 - Related resources
 - Using JWT
 - Using OAuth2
 - Common issues
- Salesmate credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API token
- SearXNG credentials
 - Supported authentication methods
 - Related resources
 - Using API URL
- SeaTable credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- SecurityScorecard credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Segment credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Sekoia credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Send Email credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using SMTP account
- Gmail Send Email credentials
 - Prerequisites
 - Set up the credential
- Outlook.com Send Email credentials
 - Set up the credential
 - Use an app password
- Yahoo Send Email credentials
 - Prerequisites
 - Set up the credential
- SendGrid credentials
 - Supported authentication methods
 - Related resources
 - Using API key
- Sendy credentials
 - Prerequisites
 - Supported authentication methods

- Related resources
- Using API Key
- Sentry.io credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API token
 - Using OAuth
 - Using Server API token
- Serp credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- ServiceNow credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using basic auth
 - Using OAuth2
- seven credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Shopify credentials
 - Supported authentication methods
 - Related resources
 - Using access token
 - Using OAuth2
 - Using API key
 - Common issues
- Shuffler credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- SIGNL4 credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using webhook secret
- Slack credentials
 - Supported authentication methods
 - Related resources
 - Using API access token
 - Using OAuth2
 - Scopes
 - Common issues
- Snowflake credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using database connection
- SolarWinds IPAM credentials
 - Supported authentication methods
 - Related resources
 - Using Username & Password
- SolarWinds Observability SaaS credentials
 - Supported authentication methods

- Related resources
- Using API Token
- Splunk credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API auth token
 - Required capabilities
- Spotify credentials
 - Supported authentication methods
 - Related resources
 - Using OAuth2
- SSH credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using password
 - Using private key
- Stackby credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Storyblok credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using Content API key
 - Using Management API key
- Strapi credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API user account
 - Using API token
- Strava credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using OAuth2
- Stripe credentials
 - Supported authentication methods
 - Related resources
 - Using API key
 - Test mode and live mode
 - Key prefixes
- Supabase credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using access token
- SurveyMonkey credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API access token
 - Using OAuth
 - Required app scopes
- SyncroMSP credentials
 - Prerequisites

- Supported authentication methods
- Related resources
- Using API key
- Sysdig Management credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API access key
- Taiga credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using basic auth
- Tapfiliate credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Telegram credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API bot access token
- TheHive credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- TheHive 5 credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- TimescaleDB credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using database connection
- Todoist credentials
 - Supported authentication methods
 - Related resources
 - Using API key
 - Using OAuth2
- Toggl credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using basic auth
- TOTP credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using secret and label
- Travis CI credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API token
- Trellix ePO credentials
 - Prerequisites

- Supported authentication methods
- Related resources
- Using basic auth
- Trello credentials
 - Supported authentication methods
 - Related resources
 - Using API key
- Twake credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using Cloud API key
 - Using Server API key
- Twilio credentials
 - Supported authentication methods
 - Related resources
 - Using Auth Token
 - Using API key
- Twist credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using OAuth2
- Typeform credentials
 - Supported authentication methods
 - Related resources
 - Using API token
 - Using OAuth2
- Unleashed Software credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- UpLead credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- uProc credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API Key
- UptimeRobot credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- urlscan.io credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Venafi TLS Protect Cloud credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Venafi TLS Protect Datacenter credentials
 - Prerequisites

- Supported authentication methods
- Related resources
- Using API integration
- Vercel AI Gateway credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
 - Using OIDC token
- Vero credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API auth token
- VirusTotal credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Vonage credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Weavate credentials
 - Supported authentication methods
 - Related resources
 - Using API key
- Webex by Cisco credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using OAuth2
- Webflow credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API access token
 - Using OAuth2
- Webhook credentials
 - Prerequisites
 - Supported authentication methods
 - Using JWT auth
- Wekan credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using basic auth
- WhatsApp Business Cloud credentials
 - Requirements
 - Supported authentication methods
 - Related resources
 - Using API key
 - Using OAuth2
- Wise credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API token
 - Add a private key

- Wolfram|Alpha credentials
 - Supported authentication methods
 - Related resources
 - Using API key
 - Resolve Forbidden connection error
- WooCommerce credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
 - Resolve "Consumer key is missing" error
- WordPress credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using basic auth
- Workable credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Wufoo credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- X (formerly Twitter) credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using OAuth2
 - X rate limits
- xAI credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Xata credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Xero credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using OAuth2
- Yourls credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Zabbix credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Zammad credentials
 - Prerequisites
 - Supported authentication methods

- Related resources
- Using basic auth
- Using token auth
- Zendesk credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API token
 - Using OAuth2
- Zep credentials
 - Supported authentication methods
 - Related resources
 - Using API key
- Zoho credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using OAuth2
- Zoom credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API JWT token
 - Using OAuth2
- Zscaler ZIA credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using basic auth and API key combo
- Zulip credentials
 - Prerequisites
 - Supported authentication methods
 - Related resources
 - Using API key
- Custom API operations
 - Predefined credential types
- Handling API rate limits
 - Identify rate limit issues
 - Handle rate limits for integrations
 - Handle rate limits in the HTTP Request node
- Install and manage community nodes
- Install verified community nodes in the n8n app
 - Install a community node
 - Uninstall a community node
- Install community nodes from npm in the n8n app
 - Install a community node
 - Uninstall a community node
 - Upgrade a community node
 - Downgrade a community node
- Manually install community nodes from npm
 - Install a community node
 - Uninstall a community node
 - Upgrade a community node
- Risks when using community nodes
 - Report bad community nodes
 - Disable community nodes
- n8n community node blocklist
- Using community nodes
 - Adding community nodes to your workflow
 - Community nodes with duplicate names

- Troubleshooting and errors
 - Error: Missing packages
 - Prevent loading community nodes on n8n cloud
- Building community nodes
- Creating nodes
 - Prerequisites
- Plan a node
- Node types: Trigger and Action
 - Trigger nodes
 - Action nodes
- Choose your node building approach
 - Data handling differences
 - Syntax differences
- Design your node's user interface
 - Design guidance
 - Standards
- Node file structure
 - Required files and directories
 - Modular structure
 - Versioning
 - Decide how many nodes to include in a package
 - A best-practice example for programmatic nodes
- Build a node
 - Set up your development environment
 - Requirements
 - Editor setup
 - Using the n8n-node tool
 - Get n8n-node
 - Command overview
 - Creating a new node
 - Building your node
 - Lint your node
 - Testing your node in n8n
 - Release your node
 - Build a declarative-style node
 - Prerequisites
 - Build your node
 - Test your node
 - Next steps
 - Build a programmatic-style node
 - Prerequisites
 - Build your node
 - Test your node
 - Next steps
- Node building reference
- Node user interface elements
 - String
 - Number
 - Collection
 - DateTime
 - Boolean
 - Color
 - Options
 - Multi-options
 - Filter
 - Assignment collection (drag and drop)
 - Fixed collection
 - Resource locator
 - Resource mapper
 - JSON

- HTML
- Notice
- Hints
- Code standards
 - Use the linter
 - Use the starter
 - Write in TypeScript
 - Detailed guidelines for writing a node
 - Detailed guidelines for writing a programmatic-style node
- Error handling in n8n nodes
 - NodeApiError
 - NodeOperationError
- Node versioning
 - Light versioning
 - Full versioning
- Node file structure
 - Required files and directories
 - Modular structure
 - Versioning
 - Decide how many nodes to include in a package
 - A best-practice example for programmatic nodes
- Node base file
 - Structure of the node base file
 - Outline structure for a declarative-style node
 - Outline structure for a programmatic-style node
- Standard parameters
 - displayName
 - name
 - icon
 - group
 - description
 - defaults
 - forceInputNodeExecution
 - inputs
 - outputs
 - requiredInputs
 - credentials
 - requestDefaults
 - properties
- Declarative-style parameters
 - methods and loadOptions
 - routing
 - version
- Programmatic-style parameters
 - defaultVersion
 - methods and loadOptions
 - version
- Programmatic-style execute() method
- Node codex files
 - Node categories
- Credentials file
 - Structure of the credentials file
 - Parameters
- HTTP request helper for node builders
 - Usage
 - Example
 - Deprecation of the previous helper
 - Migration guide to the new helper
- Item linking
- UX guidelines for community nodes

- Credentials
- Node structure
- Node functionality
- Copy
- Errors
- Community node verification guidelines
 - Use the n8n-node tool
 - Package source verification
 - No external dependencies
 - Proper documentation
 - No access to environment variables or file system
 - Follow n8n best practices
 - Use English language only
- Test a node
- Run your node locally
- n8n node linter
 - Setup
 - Usage
- Troubleshooting
 - Credentials
 - Editor UI
- Deploy a node
- Submit community nodes
- Install private nodes
 - Install your node in a Docker n8n instance
 - Install your node in a global n8n instance
- Self-hosting n8n
- Community Edition Features
 - Registered Community Edition
- npm
 - Try n8n with npx
 - Install globally with npm
 - Updating
 - Reverting an upgrade
 - Windows troubleshooting
- Docker Installation
 - Prerequisites
 - Starting n8n
 - Using with PostgreSQL
 - Updating
 - Next steps
- Server setups
- Hosting n8n on DigitalOcean
 - Create a Droplet
 - Log in to your Droplet and create new user
 - Clone configuration repository
 - Default folders and files
 - Set up DNS
 - Open ports
 - Configure n8n
 - The Docker Compose file
 - Configure Caddy
 - Start Docker Compose
 - Test your setup
 - Stop n8n and Caddy
 - Updating
 - Next steps
- Hosting n8n on Heroku
 - Use the deployment template to create a Heroku project
 - Changing the deployment template

Next steps

Hosting n8n on Hetzner cloud

- Create a server
- Log in to your server
- Install Docker Compose
- Clone configuration repository
- Default folders and files
- Set up DNS
- Open ports
- Configure n8n
- The Docker Compose file
- Configure Caddy
- Start Docker Compose
- Test your setup
- Stop n8n and Caddy
- Updating
- Next steps

Hosting n8n on Amazon Web Services

- Hosting options
- Prerequisites
- Create a cluster
- Clone configuration repository
- Configure Postgres
- Configure n8n
- Deployments
- Services
- Send to Kubernetes cluster
- Set up DNS
- Delete resources
- Next steps

Hosting n8n on Azure

- Prerequisites
- Hosting options
- Open the Azure Kubernetes Service
- Create a cluster
- Set Kubectl context
- Clone configuration repository
- Configure Postgres
- Configure n8n
- Deployments
- Services
- Send to Kubernetes cluster
- Set up DNS
- Delete resources
- Next steps

Hosting n8n on Google Cloud Run

- Before you begin: get a Google Cloud project
- Easy mode
- Durable mode
- Enable APIs and set env vars
- Setup your Postgres database
- Store sensitive data in Secret Manager
- Create a service account for Cloud Run
- Deploy the Cloud Run service
- Troubleshooting
- (Optional) Enabling Google Workspace services as n8n tools

Hosting n8n on Google Kubernetes Engine

- Prerequisites
- Create project
- Enable the Kubernetes Engine API

Create a cluster
Set Kubectl context
Clone configuration repository
Configure Postgres
Configure n8n
Deployments
Services
Send to Kubernetes cluster
Set up DNS
Delete resources
Next steps
Docker-Compose
1. Install Docker and Docker Compose
2. Optional: Non-root user access
3. DNS setup
4. Create an .env file
The subdomain to serve from
The above example serve n8n at: <https://n8n.example.com>
Optional timezone to set which gets used by Cron and other scheduling nodes
New York is the default value if not set
The email address to use for the TLS/SSL certificate creation
6. Create Docker Compose file
8. Done
Next steps
Update self-hosted n8n
Environment variables overview
AI Assistant environment variables
Binary data environment variables
Credentials environment variables
Database environment variables
 PostgreSQL
 SQLite
Deployment environment variables
Endpoints environment variables
Executions environment variables
External data storage environment variables
External hooks environment variables
External secrets environment variables
Insights environment variables
Logs environment variables
 n8n logs
 Log streaming
License environment variables
Nodes environment variables
Queue mode environment variables
 Multi-main setup
Security environment variables
Source control environment variables
Task runner environment variables
 n8n instance environment variables
 Task runner launcher environment variables
 Task runner environment variables (all languages)
 Task runner environment variables (JavaScript)
 Task runner environment variables (Python)
Timezone and localization environment variables
User management SMTP, and two-factor authentication environment variables
Workflows environment variables
Configuration

- Set environment variables by command line
- Docker Compose file
- Keeping sensitive data in separate files
- Configuration examples
- Isolate n8n
 - Configure the Base URL for n8n's front end access
 - Configure n8n to use your own certificate authority or self-signed certificate
 - Docker
 - Certificate requirements for Custom Trust Store
 - Set a custom encryption key
 - Configure workflow timeout settings
 - Specify location for your custom nodes
 - Enable modules in Code node
 - Set the self-hosted instance timezone
 - Specify user folder path
 - Configure n8n webhooks with reverse proxy
 - Enable Prometheus metrics
 - Queue metrics
- Supported databases
 - Database type by n8n installation
 - Shared settings
 - PostgresDB
 - SQLite
- Task runners
 - How it works
 - Task runner modes
 - Setting up external mode
 - Adding extra dependencies
- Configure self-hosted n8n for user management
 - Setup
- Logging in n8n
 - Setup
 - Development
 - Front-end logs
- Monitoring
 - healthz and healthz/readiness
 - metrics
 - Enable metrics and healthz for self-hosted n8n
- Security audit
 - Run an audit
 - Report contents
- Scaling n8n
- Performance and benchmarking
 - Performance factors
 - Run your own benchmarking
 - Example: Single instance performance
 - Example: Multi-instance performance
- Queue mode
 - How it works
 - Configuring workers
 - Running n8n with queues
 - Webhook processors
 - Configure worker concurrency
 - Concurrency and scaling recommendations
 - Multi-main setup
- Self-hosted concurrency control
 - Comparison to queue mode
- Execution data
 - Reduce saved data

- Enable executions pruning
- Binary data
 - Enable filesystem mode
 - Binary data pruning
- External storage
 - Storing n8n's binary data in S3
- Memory-related errors
 - Identifying out of memory situations
 - Typical causes
 - Avoiding out of memory situations
- Securing n8n
- Set up SSL
 - Use a reverse proxy (recommended)
 - Pass certificates into n8n directly
- Set up Single Sign-On (SSO)
- Security audit
 - Run an audit
 - Report contents
- Disable the public REST API
 - Disable the API playground
 - Related resources
- Data collection
 - Collected data
 - How collection works
 - Opting out of data collection
 - Disable all connection to n8n servers
 - Related resources
- Block access to nodes
 - Exclude nodes
 - Suggested nodes to block
 - Related resources
- Hardening task runners
 - Run task runners as sidecars in external mode
- Restrict account registration to email-verified users
 - Prerequisites
 - How to restrict account registration
- Self-hosted AI Starter Kit
 - What's included
 - What you can build
 - Get the kit
- Architecture
- Database structure
 - Database and query technology
 - Tables
 - Entity Relationship Diagram (ERD)
- CLI commands for n8n
 - Running CLI commands
 - Start a workflow
 - Change the active status of a workflow
 - Export entities
 - Export workflows and credentials
 - Import entities
 - Import workflows and credentials
 - License
 - User management
 - Uninstall community nodes and credentials
 - Security audit
- Code in n8n
 - Code in your workflows
 - Other technical resources

- Expressions
 - Writing expressions
 - Common issues
- Using the Code node
- AI coding with GPT
 - Use AI in the Code node
 - Usage limits
 - Feature limits
 - Writing good prompts
 - Fixing the code
- Built-in methods and variables
 - Current node input
 - Output of other nodes
 - Built-in date and time methods
 - JMESPath method
 - HTTP node variables
 - LangChain Code node methods
 - n8n metadata
 - Convenience methods
 - Data transformation functions
 - Usage
 - Arrays
 - Booleans
 - Dates
 - Numbers
 - Objects
 - Strings
 - Custom variables
 - Create variables
 - Edit and delete variables
 - Use variables in workflows
- Date and time with Luxon
 - Date and time behavior in n8n
 - Setting the timezone in n8n
 - Common tasks
- Query JSON with JMESPath
 - The `jmespath()` method
 - Common tasks
- Examples using n8n's built-in methods and variables
 - Related resources
- execution
 - `execution.id`
 - `execution.resumeUrl`
 - `execution.customData`
- getWorkflowStaticData(type)
 - Templates and examples
- Retrieve linked items from earlier in the workflow
 - (`<node-name>`).`all(branchIndex?: number, runIndex?: number)`
 - Getting items
 - Accessing item data
- vars
- Expressions cookbook
 - Related resources
- Check incoming data
- Expressions common issues
 - The 'JSON Output' in item 0 contains invalid JSON
 - Can't get data for expression
 - Invalid syntax
- Code node cookbook
 - Related resources

Get number of items returned by the previous node
Get the binary data buffer
Output to the browser console with `console.log()` or `print()` in the Code node
 `console.log` (JavaScript)
 `print` (Python)
Examples using n8n's HTTP Request node
 Related resources
Pagination in the HTTP Request node
 Enable pagination
 Use a URL from the response to get the next page using `$response`
 Get the next page by number using `$pageCount`
 Navigate pagination through body parameters
 Set the page size in the query
Advanced AI
 Related resources
AI Workflow Builder
 Working with the builder
 Understanding credits
 AI model and data handling
Accessing n8n MCP server
 Difference between instance-level MCP access and MCP Server
Trigger node
 Enabling MCP access
 Setting up MCP authentication
 Exposing workflows to MCP clients
 Executing workflows through MCP clients
 Examples
 Troubleshooting
Build an AI chat agent with n8n
 AI concepts in n8n
 1. Create a new workflow
 2. Add a trigger node
 3. Add an AI Agent Node
 4. Configure the node
 5. Add credentials (if needed)
 6. Test the node
 7. Changing the prompt
 8. Adding persistence
 9. Saving the workflow
 Congratulations!
 Next steps
RAG in n8n
 What is RAG
 What is a vector store?
 How to use RAG in n8n
 FAQs
LangChain in n8n
LangChain concepts in n8n
 Trigger nodes
 Cluster nodes
LangChain learning resources
Use LangSmith with n8n
 Connect your n8n instance to LangSmith
Overview
 What are evaluations?
 Why is evaluation needed?
 Two types of evaluation
 Learn more
Light evaluations

- What are light evaluations?
- How it works
- Metric-based evaluations
 - How it works
- Tips and common issues
 - Combining multiple triggers
 - Avoiding evaluation breaking the chat
 - Accessing tool data when calculating metrics
 - Multiple evaluations in the same workflow
 - Dealing with inconsistent results
- Advanced AI examples and concepts
- What's a chain in AI?
 - Chains in n8n
- What's an agent in AI?
 - Agents in n8n
- Demonstration of key differences between agents and chains
 - Key features
 - Using the example
- What's memory in AI?
 - AI memory in n8n
- What's a tool in AI?
 - AI tools in n8n
- Chat with a Google Sheet using AI
 - Key features
 - Using the example
- Call an API to fetch data
 - Key features
 - Using the example
- Have a human fallback for AI workflows
 - Key features
 - Using the example
- Let AI specify the tool parameters
 - Let the model fill in the parameter
 - Use the `$fromAI()` function
- What are vector databases?
 - A simplified example
 - Demonstrating the power of similarity search
 - Embeddings, retrievers, text splitters, and document loaders
- Populate a Pinecone vector database from a website
 - Key features
 - Using the example
- n8n public REST API
 - Learn about REST APIs
- API authentication
 - API Scopes
 - Create an API key
 - Call the API using your key
 - Delete an API key
- API pagination
- Using an API playground
 - Documentation playground
 - Built-in playground
- n8n Embed
 - Support
 - Russia and Belarus
- Prerequisites
 - CPU considerations
 - Database considerations
 - Memory considerations
- Deployment

- User data
- Backups
- Restarting
- Configuration
 - Authentication
 - Environment variables
 - Backend hooks
 - Frontend external hooks
- Workflow management in Embed
 - Workflow per user
 - Single workflow
- Workflow templates
 - Disable workflow templates
 - Use your own workflow templates library
 - Add your workflows to the n8n library
- White labelling
 - Prerequisites
 - Theme colors
 - Theme logos
 - Text localization

n8n Documentation

Exported from docs.n8n.io

Welcome to n8n Docs

This is the documentation for [n8n](#), a [fair-code](#) licensed workflow automation tool that combines AI capabilities with business process automation.

It covers everything from setup to usage and development. It's a work in progress and all [contributions](#) are welcome.

Where to start

- **Quickstarts**

Jump in with n8n's quickstart guides.

[:octicons-arrow-right-24:](#) Try it out

- **Choose the right n8n for you**

Cloud, npm, self-host . . .

[:octicons-arrow-right-24:](#) Options

- **Explore integrations**

Browse n8n's integrations library.

[:octicons-arrow-right-24:](#) Find your apps

- **Build AI functionality**

n8n supports building AI functionality and tools.

 [Advanced AI](#)

About n8n

n8n (pronounced n-eight-n) helps you to connect any app with an API with any other, and manipulate its data with little or no code.

- Customizable: highly flexible workflows and the option to build custom nodes.
- Convenient: use the npm or Docker to try out n8n, or the Cloud hosting option if you want us to handle the infrastructure.
- Privacy-focused: self-host n8n for privacy and security.

This guide outlines a series of tutorials and resources designed to get you started with n8n.

It's not necessary to complete all items listed to start using n8n. Use this as a reference to navigate to the most relevant parts of the documentation and other resources according to your needs.

Join the community

n8n has an active community where you can get and offer help. Connect, share, and learn with other n8n users:

- [Ask questions](#) and [make feature requests](#) in the Community Forum.
- [Report bugs](#) and [contribute](#) on GitHub.

Set up your n8n

If you don't have an account yet, sign up to a [free trial on n8n Cloud](#) or install n8n's community edition with [Docker](#) (recommended) or [npm](#). See [Choose your n8n](#) for more details.

Try it out

Start with the quickstart guides to help you get up and running with building basic workflows.

- [A very quick quickstart](#)
- [A longer introduction](#)
- [Build an AI workflow in n8n](#)

Structured Courses

n8n offers two sets of courses.

Video courses

Learn key concepts and n8n features, while building examples as you go.

- The [Beginner](#) course covers the basics of n8n.
- The [Advanced](#) course covers more complex workflows, more technical nodes, and enterprise features

Text courses

Build more complex workflows while learning key concepts along the way. Earn a badge and an avatar in your community profile.

- [Level 1: Beginner Course](#)
- [Level 2: Intermediate Course](#)

Self-hosting n8n

Explore various [self-hosting options in n8n](#). If you're not sure where to start, these are two popular options:

- [Hosting n8n on DigitalOcean](#)
- [Hosting n8n on Amazon Web Services](#)

Build a node

If you can't find a node for a specific app or a service, you can build a node yourself and share with the community. See what others have built on [npm website](#).

- [Build a declarative-style node](#)
- [Learn how to build your own n8n nodes \(Youtube Video\)](#)

Stay updated

- Follow new features and bug fixes in the [Release Notes](#)
 - Follow n8n on socials: [Twitter/X](#), [Discord](#), [LinkedIn](#), [YouTube](#)
-

Choose your n8n

This section contains information on n8n's range of platforms, pricing plans, and licenses.

Platforms

There are different ways to set up n8n depending on how you intend to use it:

- [n8n Cloud](#): hosted solution, no need to install anything.
- [Self-host](#): recommended method for production or customized use cases.
 - [npm](#)
 - [Docker](#)
 - [Server setup guides](#) for popular platforms
- [Embed](#): n8n Embed allows you to white label n8n and build it into your own product. Contact n8n on the [Embed website](#) for pricing

and support.

-8<- “_snippets/self-hosting/warning.md”

Licenses

n8n's [Sustainable Use License](#) and [n8n Enterprise License](#) are based on the [fair-code](#) model.

For a detailed explanation of the license, refer to [Sustainable Use License](#).

Free versions

n8n offers the following free options:

- A free trial of Cloud
- A free self-hosted community edition for self-hosted users

Paid versions

n8n has two paid versions:

- n8n Cloud: choose from a range of paid plans to suit your usage and feature needs.
- Self-hosted: there are both free and paid versions of self-hosted.

For details of the Cloud plans and contact details for Enterprise Self-hosted, refer to [Pricing](#) on the n8n website.

Try it out

The best way to learn n8n is by using our tutorials to get familiar with the user interface and the many different types of nodes and integrations available. Here is a selection of material to get you started:

- Looking for a quick introduction? Check out the [“First Workflow” tutorial](#).
 - Interested in what you could do with AI? Find out [how to build an AI chat agent with n8n](#).
 - Prefer to work through extensive examples? Maybe the [courses](#) are for you.
-

The very quick quickstart

This quickstart gets you started using n8n as quickly as possible. Its allows you to try out the UI and introduces two key features: [workflow templates](#) and [expressions](#). It doesn't include detailed explanations or explore concepts in-depth.

In this tutorial, you will:

- Load a [workflow](#) from the workflow templates library
- Add a node and configure it using expressions
- Run your first workflow

Step one: Open a workflow template and sign up for n8n Cloud

n8n provides a quickstart template using training nodes. You can use this to work with fake data and avoid setting up [credentials](#).

This quickstart uses [n8n Cloud](#). A free trial is available for new users.

1. Go to [Templates](#) | [Very quick quickstart](#).
2. Select **Use for free** to view the options for using the template.
3. Select **Get started free with n8n cloud** to sign up for a new Cloud instance.

This workflow:

1. Gets example data from the [Customer Datastore](#) node.
2. Uses the [Edit Fields](#) node to extract only the desired data and assigns that data to variables. In this example, you map the customer name, ID, and description.

The individual pieces in an n8n workflow are called [nodes](#). Double click a node to explore its settings and how it processes data.

Step two: Run the workflow

Select **Execute Workflow**. This runs the workflow, loading the data from the Customer Datastore node, then transforming it with Edit Fields. You need this data available in the workflow so that you can work with it in the next step.

Step three: Add a node

Add a third node to message each customer and tell them their description. Use the Customer Messenger node to send a message to fake recipients.

1. Select the **Add node** Image: Add node icon connector on the Edit Fields node.
2. Search for **Customer Messenger**. n8n shows a list of nodes that match the search.
3. Select **Customer Messenger (n8n training)** to add the node to the [canvas](#). n8n opens the node automatically.
4. Use [expressions](#) to map in the **Customer ID** and create the **Message**:
 1. In the **INPUT** panel select the **Schema** tab.
 2. Drag **Edit Fields1 > customer_id** into the **Customer ID** field in the node settings.
 3. Hover over **Message**. Select the **Expression** tab, then select the expand button Image: Add node icon to open the full expressions editor.
 4. Copy this expression into the editor: `Hi {{ $json.customer_name }}`. Your description is: `{{ $json.customer_description }}`

5. Close the expressions editor, then close the **Customer Messenger** node by clicking outside the node or selecting **Back to canvas**.
6. Select **Execute Workflow**. n8n runs the workflow.

The complete workflow should look like this:

Next steps

- Read n8n's [longer try it out tutorial](#) for a more complex workflow, and an introduction to more features and n8n concepts.
 - Take the [text courses](#) or [video courses](#).
-

Your first workflow

This guide will show you how to construct a [workflow](#) in n8n, explaining key concepts along the way. You will:

- Create a workflow from scratch.
- Understand key concepts and skills, including:
 - Starting workflows with trigger nodes
 - Configuring [credentials](#)
 - Processing data
 - Representing logic in an n8n workflow
 - Using [expressions](#)

[Image: "Screenshot of the completed workflow"]

This quickstart uses [n8n Cloud](#), which is recommended for new users. A free trial is available - if you haven't already done so, [sign up](#) for an account now.

Step one: Create a new workflow

When you open n8n, you'll see either:

- A window with a welcome message and two large buttons: Choose **Start from Scratch** to create a new workflow.
- The **Workflows** list on the [Overview](#) page. Select the **Create Workflow** to create a new workflow.

Step two: Add a trigger node

n8n provides two ways to start a workflow:

- Manually, by selecting **Execute Workflow**.
- Automatically, using a trigger node as the first node. The trigger node runs the workflow in response to an external event, or based on your settings.

For this tutorial, we'll use the [Schedule trigger](#). This allows you to run the workflow on a schedule:

1. Select **Add first step**.
2. Search for **Schedule**. n8n shows a list of nodes that match the

- search.
3. Select **Schedule Trigger** to add the node to the canvas. n8n opens the node.
 4. For **Trigger Interval**, select **Weeks**.
 5. For **Weeks Between Triggers**, enter 1.
 6. Enter a time and day. For this example, select **Monday** in **Trigger on Weekdays**, select **9am** in **Trigger at Hour**, and enter 0 in **Trigger at Minute**.
 7. Close the node details view to return to the canvas.

Step three: Add the NASA node and set up credentials

The [NASA node](#) interacts with NASA's [public APIs](#) to fetch useful data. We will use the real-time data from the API to find solar events.

??? explanation "Credentials" Credentials are private pieces of information issued by apps and services to authenticate you as a user and allow you to connect and share information between the app or service and the n8n node. The type of information required varies depending on the app/service concerned. You should be careful about sharing or revealing the credentials outside of n8n.

1. Select the **Add node** Image: Add node icon connector on the Schedule Trigger node.
2. Search for **NASA**. n8n shows a list of nodes that match the search.
3. Select **NASA** to view a list of operations.
4. Search for and select **Get a DONKI solar flare**. This operation returns a report about recent solar flares. When you select the operation, n8n adds the node to the canvas and opens it.
5. To access the NASA APIs, you need to set up credentials:
 1. Select the **Credential for NASA API** dropdown.
 2. Select **Create new credential**. n8n opens the credentials view.
 3. Go to [NASA APIs](#) and fill out the form from the **Generate API Key** link. The NASA site generates the key and emails it to the address you entered.
 4. Check your email account for the API key. Copy the key, and paste it into **API Key** in n8n.
 5. Select **Save**.
 6. Close the credentials screen. n8n returns to the node. The new credentials should be automatically selected in **Credential for NASA API**.
6. By default, DONKI Solar Flare provides data for the past 30 days. To limit it to just the last week, use **Additional Fields**:
 1. Select **Add field**.
 2. Select **Start date**.
 3. To get a report starting from a week ago, you can use an expression: next to **Start date**, select the **Expression** tab, then select the expand button Image: Add node icon to open the full expressions editor.
 4. In the **Expression** field, enter the following expression:

```
{{ $today.minus(7, 'days') }}
```

This generates a date in the correct format, seven days before the current date.

[Image: image showing the expression above generating a date]

??? explanation “Date and time formats in n8n...” n8n uses Luxon to work with date and time, and also provides two variables for convenience: `$now` and `$today`. For more information, refer to [Expressions > Luxon](#).

7. Close the **Edit Expression** modal to return to the NASA node.
8. You can now check that the node is working and returning the expected date: select **Execute step** to run the node manually. n8n calls the NASA API and displays details of solar flares in the past seven days in the **OUTPUT** section.
9. Close the NASA node to return to the workflow canvas.

Step four: Add logic with the If node

n8n supports complex logic in workflows. In this tutorial we will use the **If node** to create two branches that each generate a report from the NASA data. Solar flares have five possible classifications; we will add logic that sends a report with the lower classifications to one output, and the higher classifications to another.

Add the If node:

1. Select the **Add node** Image: Add node icon connector on the NASA node.
2. Search for **If**. n8n shows a list of nodes that match the search.
3. Select **If** to add the node to the canvas. n8n opens the node.
4. You need to check the value of the `classType` property in the NASA data. To do this:
 1. Drag **classType** into **Value 1**.
 2. Change the comparison operation to **String > Contains**.
 3. In **Value 2**, enter **X**. This is the highest classification of solar flare. In the next step, you will create two reports: one for X class solar flares, and one for all the smaller solar flares.
5. You can now check that the node is working and returning the expected date: select **Execute step** to run the node manually. n8n tests the data against the condition, and shows which results match true or false in the **OUTPUT** panel.
6. Once you are happy the node will return some events, you can close the node to return to the canvas.

Step five: Output data from your workflow

The last step of the workflow is to send the two reports about solar flares. For this example, you'll send data to [Postbin](#). Postbin is a service that receives data and displays it on a temporary web page.

1. On the If node, select the **Add node** Image: Add node icon connector labeled **true**.
2. Search for **PostBin**. n8n shows a list of nodes that match the search.
3. Select **PostBin**.
4. Select **Send a request**. n8n adds the node to the canvas and opens it.
5. Go to [Postbin](#) and select **Create Bin**. Leave the tab open so you can come back to it when testing the workflow.
6. Copy the bin ID. It looks similar to 1651063625300-2016451240051.
7. In n8n, paste your Postbin ID into **Bin ID**.
8. Now, configure the data to send to Postbin. Next to **Bin Content**, select the **Expression** tab (you will need to mouse-over the **Bin Content** for the tab to appear), then select the expand button Image: Add node icon to open the full expressions editor.
9. You can now click and drag the correct field from the If Node output into the expressions editor to automatically create a reference for this label. In this case the input we want is 'classType'.
10. Once dropped into the expressions editor it will transform into this reference: `{{$json["classType"]}}`. Add a message to it, so that the full expression is:

```
There was a solar flare of class  {{$json["classType"]}}
```

[Image: image showing the expression above generating output]

11. Close the expressions editor to return to the node.
12. Close the Postbin node to return to the canvas.
13. Add another Postbin node, to handle the **false** output path from the If node:
 1. Hover over the Postbin node, then select **Node context menu** Image: Node context menu icon > **Duplicate node** to duplicate the first Postbin node.
 2. Drag the **false** connector from the If node to the left side of the new Postbin node.

Step six: Test the workflow

1. You can now test the entire workflow. Select **Execute Workflow**. n8n runs the workflow, showing each stage in progress.
2. Go back to your Postbin bin. Refresh the page to see the output.
3. If you want to use this workflow (in other words, if you want it to run once a week automatically), you need to publish it by clicking **Publish**.

Congratulations

You now have a fully functioning workflow that does something useful! It should look something like this:

Along the way you have discovered:

- How to find the nodes you want and join them together
- How to use expressions to manipulate data
- How to create credentials and attach them to nodes
- How to use logic in your workflows

There are plenty of things you could add to this (perhaps add some more credentials and a node to send you an email of the results), or maybe you have a specific project in mind. Whatever your next steps, the resources linked below should help.

Next steps

- Interested in what you could do with AI? Find out [how to build an AI chat agent with n8n](#).
 - Take n8n's [text courses](#) or [video courses](#).
 - Explore more examples in [workflow templates](#).
-

Video courses

n8n provides two video courses on YouTube.

For support, join the [Forum](#).

Beginner

The [Beginner](#) course covers the basics of n8n:

- [Introduction and workflows](#)
- [APIs and Webhooks](#)
- [Nodes](#)
- [Data in n8n](#)
- [Core workflow concepts](#)
- [Useful nodes](#)
- [Error handling](#)
- [Debugging](#)
- [Collaboration](#)

Advanced

The [Advanced](#) course covers more complex workflows, more technical nodes, and enterprise features:

- [Introduction and complex data flows](#)
- [Advanced technical nodes](#)
- [Pinning and editing output data](#)
- [Sub-workflows](#)
- [Error workflows](#)

- [Building a full example](#)
 - [Handling files](#)
 - [Enterprise features](#)
-

Text courses

If you've found your way here, it means you're serious about your interest in automation. Maybe you're tired of manually entering data into the same spreadsheet every day, of clicking through a series of tabs and buttons for that one piece of information you need, of managing tens of different tools and systems.

Whatever the reason, one thing is clear: you shouldn't spend precious time doing things that don't spark joy or contribute to your personal and professional growth.

These tasks can and should be automated! And you don't need advanced technical knowledge or excellent coding skills to do this—with no-code tools like n8n, automation is for everyone.

Available courses

- [Level 1: Beginner course](#)
 - [Level 2: Intermediate course](#)
-

Level one: Introduction

Welcome to the **n8n Course Level 1!**

Is this course right for me?

This course introduces you to the fundamental concepts within n8n and develops your low-code automation expertise.

This course is for you if you:

- Are starting to use n8n for the first time.
- Are looking for some extra help creating your first workflow.
- Want to automate processes in your personal or working life.

This course introduces n8n concepts and demonstrates practical workflow building without assuming any prior familiarity with n8n. If you'd like to get a feel for the basics without as much explanation, consult our [quickstart guide](#).

What will I learn in this course?

We believe in learning by doing. You can expect some theoretical information about the basic concepts and components of n8n, followed by practice of building workflows step by step.

By the end of this course you will know:

- How to set up n8n and navigate the Editor UI.
- How n8n structures data.
- How to configure different node parameters and add credentials.
- When and how to use conditional logic in workflows.
- How to schedule and control workflows.
- How to import, download, and share workflows with others.

You will build two workflows:

- A two-node workflow to get articles from Hacker News
- A seven-node workflow to help your client get records from a data warehouse, filter them, make calculations, and notify team members about the results

What do I need to get started?

1. **n8n set up:** You can use [n8n Cloud](#) (or the [self-hosted version](#) if you have experience hosting services).
2. **A course user ID:** [Sign up here](#) to get your unique ID and other credentials you will need in this course (Level 1).
3. Basic knowledge of JavaScript and [APIs](#) would be helpful, but isn't necessary.
4. An [account on the n8n community forum](#) if you wish to receive a profile badge and avatar upon successful completion.

How long does the course take?

Completing the course should take around **two hours**. You don't have to complete it in one go; feel free to take breaks and resume whenever you are ready.

How do I complete the course?

There are two milestones in this course that test your knowledge of what you have learned in the lessons:

- Building the [main workflow](#)
- Passing the [quiz](#) at the end of the course

If you complete the milestones above, you will get [a badge and an avatar](#) in your forum profile. You can then share your profile and course verification ID to showcase your n8n skills to others.

[Let's get started!](#)

Navigating the Editor UI

In this lesson you will learn how to navigate the [Editor UI](#). We will walk through the [canvas](#) and show you what each icon means and where to find things you will need while building workflows in n8n.

Getting started

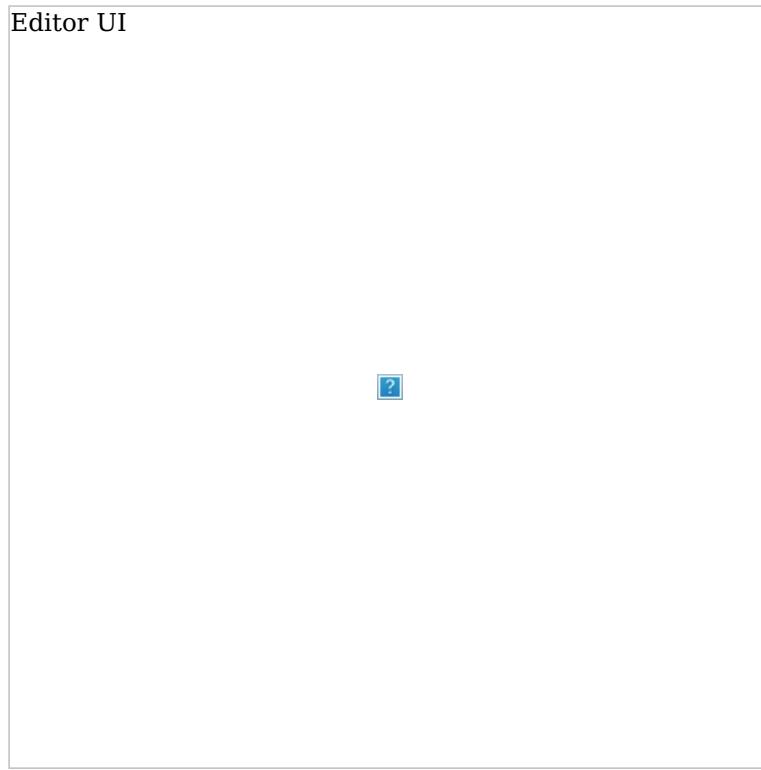
Begin by setting up n8n.

We recommend starting with [n8n Cloud](#), a hosted solution that doesn't require installation and includes a free trial.

For more details on the different ways to set up n8n, see our [platforms documentation](#).

Once you have n8n running, open the Editor UI in a browser window. Log in to your n8n instance. Select **Overview** and then **Create Workflow** to view the main canvas.

It should look like this:



Editor UI

Editor UI settings

The editor UI is the web interface where you build [workflows](#). You can access all your workflows and [credentials](#), as well as support pages, from the Editor UI.

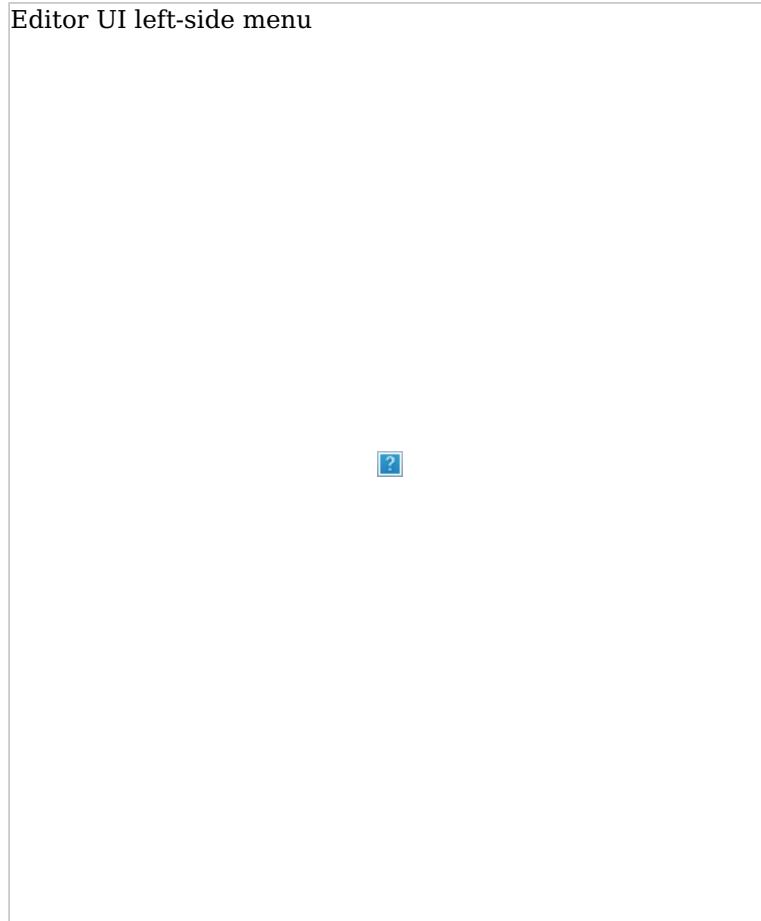
Left-side panel

On the left side of the **Editor UI**, there is a panel which contains the core functionalities and settings for managing your workflows. Expand and collapse it by selecting the small arrow icon.

The panel contains the following sections:

- **Overview:** Contains all the workflows, credentials, and executions you have access to. During this course, create new workflows here.
- **Personal:** Every user gets a default personal project. If you don't create a custom project, your workflows and credentials are stored here.
- **Projects:** Projects let you group workflows and credentials together. You can assign [roles](#) to users in a project to control what they can do. Projects aren't available on the Community edition.
- **Admin Panel:** n8n Cloud only. Access your n8n instance usage, billing, and version settings.
- **Templates:** A collection of pre-made workflows. Great place to get started with common use cases.
- **Variables:** Used to store and access fixed data across your workflows. This feature is available on the Pro and Enterprise Plans.
- **Insights:** Provides analytics and insights about your workflows.
- **Help:** Contains resources around n8n product and community.
- **What's New:** Shows the latest product updates and features.

Editor UI left-side menu



Editor UI left-side menu

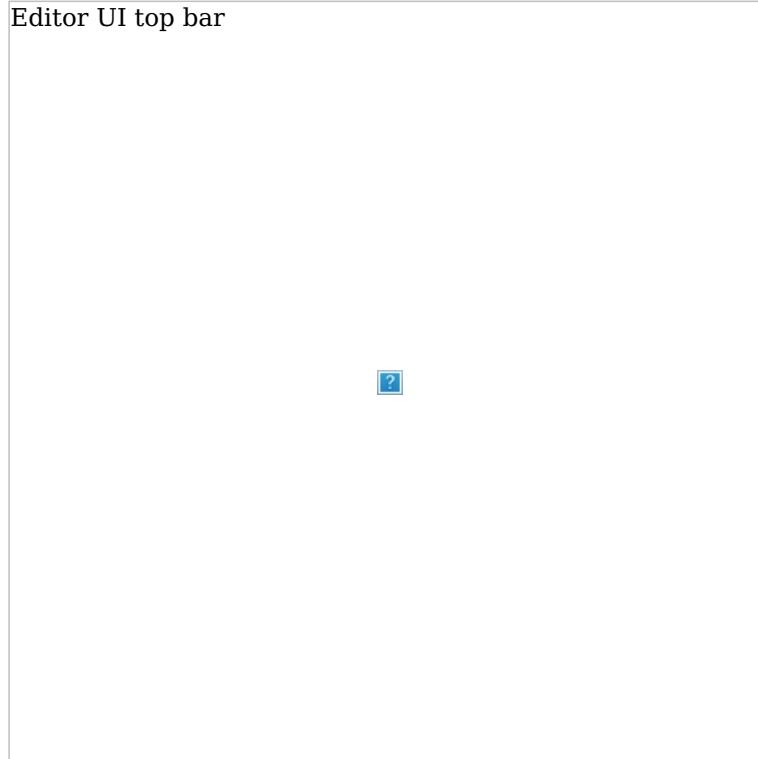
Top bar

The top bar of the **Editor UI** contains the following information:

- **Workflow Name:** By default, n8n names a new workflow as "My workflow", but you can edit the name at any time.
- **+ Add Tag:** Tags help you organise your workflows by category,

use case, or whatever is relevant for you. Tags are optional.

- **Publish:** This button publishes the current workflow. By default, workflows are not published.
- **Share:** You can share and collaborate with others on workflows on the Starter, Pro, and Enterprise plans.
- **Save:** This button saves the current workflow.
- **History:** Once you save your workflow, you can view previous versions here.

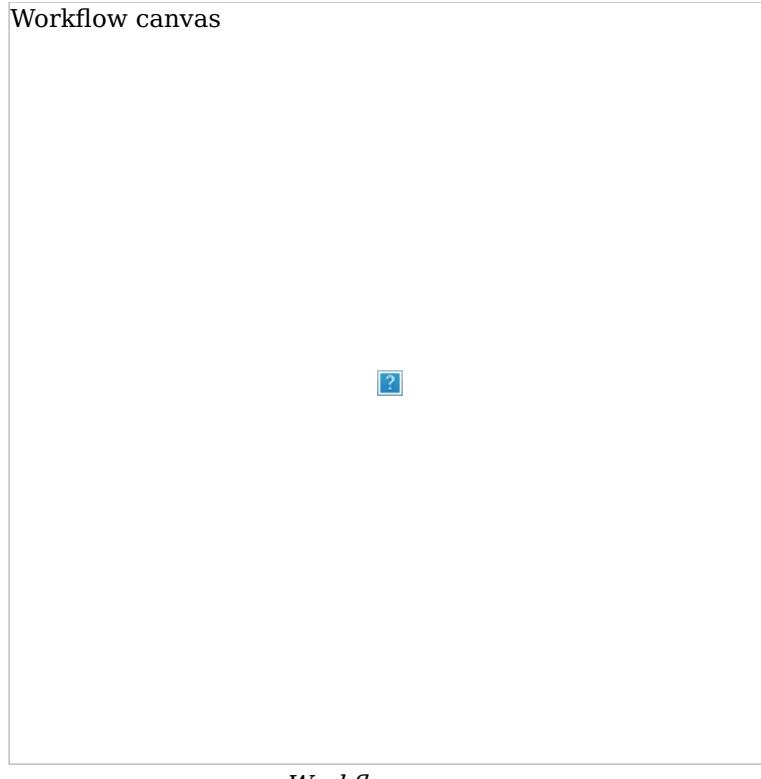


Editor UI top bar

Canvas

The **canvas** is the gray dotted grid background in the Editor UI. It displays several icons and a node with different functionalities:

- Buttons to zoom the canvas to fit the screen, zoom in or out of the canvas, reset zoom, and tidy up the nodes on screen.
- A button to **Execute workflow** once you add your first node. When you click on it, n8n executes all nodes on the canvas in sequence.
- A button with a + sign inside. This button opens the nodes panel.
- A button with a note icon inside. This button adds a sticky note to the canvas (visible when hovering on the top right + icon).
- A button labeled **Ask Assistant** appears on the right side of the canvas. You can ask the AI Assistant for help with building workflows.
- A dotted square with the text “Add first step.” This is where you add your first node.



Don't worry about workflow execution and publishing for now; we'll explain these concepts later on in the course.

Nodes

You can think of nodes as building blocks that serve different functions that, when put together, make up a functioning machine: an automated workflow.

Based on their function, n8n classifies nodes into four types:

- **App or Action Nodes** add, remove, and edit data; request and send external data; and trigger events in other systems. Refer to the [Action nodes library](#) for a full list of these nodes.
- **Trigger Nodes** start a workflow and supply the initial data. Refer to the [Trigger nodes library](#) for a list of trigger nodes.
- **Core Nodes** can be trigger or app nodes. Whereas most nodes connect to a specific external service, core nodes provide functionality such as logic, scheduling, or generic API calls. Refer to the [Core Nodes library](#) for a full list of core nodes.
- **Cluster Nodes** are node groups that work together to provide functionality in a workflow, primarily for AI workflows. Refer to [Cluster nodes](#) for more information.

Finding nodes

You can find all available nodes in the **nodes panel** on the right side of the Editor UI. There are three ways in which you can open the nodes panel:

- Click the + icon in the top right corner of the canvas.
- Click the + icon on the right side of an existing node on the canvas (the node to which you want to add another one).
- Click the ++tab++ key on your keyboard.



In the nodes panel, notice that when adding your first node, you will see the different trigger node categories. After you have added your trigger node, you'll see that the nodes panel changes to show Advanced AI, Actions in an App, Data transformation, Flow, Core, and Human in the loop nodes.

If you want to find a specific node, use the search input at the top of the nodes panel.

Adding nodes

There are two ways to add nodes to your canvas:

- Select the node you want in the nodes panel. The new node will automatically connect to the selected node on the canvas.
- Drag and drop the node from the nodes panel to the canvas.

Node buttons

If you hover on a node, you'll notice that three icons appear on top:

- Execute the node (Play icon)
- Deactivate/Activate the node (Power icon)
- Delete the node (Trash icon)

There will also be an ellipsis icon, which opens a context menu containing other node options.

Summary

In this lesson you learned how to navigate the Editor UI, what the icons mean, how to access the left-side and node panels, and how to add nodes to the canvas.

In the next lesson, you will build a mini-workflow to put into practice what you've learned so far.

Building a Mini-workflow

In this lesson, you will build a small workflow that gets 10 articles about automation from Hacker News. The process consists of five steps:

1. Add a Manual Trigger node
2. Add the Hacker News node
3. Configure the Hacker News node
4. Execute the node
5. Save the workflow

The finished workflow will look like this:

1. Add a Manual Trigger node

Open the nodes panel (reminder: you can open this by selecting the **+** icon in the top right corner of the canvas or selecting **++tab++** on your keyboard).

Then:

1. Search for the **Manual Trigger** node.
2. Select it when it appears in the search.

This will add the Manual Trigger node to your canvas, which allows you to run the workflow at any time by selecting the **Execute workflow** button.

2. Add the Hacker News node

Select the **+** icon to the right of the Manual Trigger node to open the nodes panel.

Then:

1. Search for the **Hacker News** node.
2. Select it when it appears in the search.
3. In the **Actions** section, select **Get many items**.

n8n adds the node to your canvas and the node window opens to display its configuration details.

3. Configure the Hacker News node

When you add a new node to the Editor UI, the node is automatically activated. The node details will open in a window with several options:

- **Parameters:** Adjust parameters to refine and control the node's functionality.
- **Settings:** Adjust settings to control the node's design and executions.
- **Docs:** Open the n8n documentation for this node in a new window.

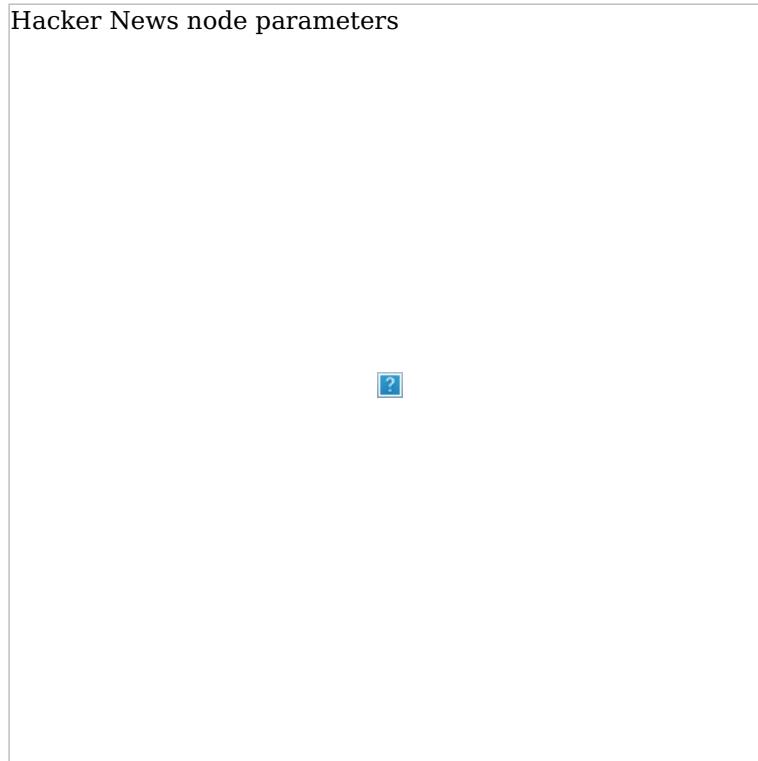
Parameters

We need to configure several parameters for the Hacker News node to make it work:

- **Resource:** All
This resource selects all data records (articles).
- **Operation:** Get Many
This operation fetches all the selected articles.
- **Limit:** 10
This parameter sets a limit to the number of results the Get Many operation returns.
- **Additional Fields > Add Field > Keyword:** automation
Additional fields are options that you can add to certain nodes to

make your request more specific or filter the results. For this example, we want to get only articles that include the keyword “automation.”

The configuration of the parameters for the Hacker News node should now look like this:



Hacker News node parameters

Settings

The **Settings** section includes several options for node design and executions. In this case, we'll configure only the final two settings, which set the node's appearance in the Editor UI canvas.

In the Hacker News node Settings, edit:

- **Notes:** Get the 10 latest articles.
- **Display note in flow?:** toggle to true
This option will display the Note under the node in the canvas.

The configuration of the settings for the Hacker News node should now look like this:

Hacker News node settings



Hacker News node settings

4. Execute the node

Select the **Execute step** button in the node details window. You should see 10 results in the Output **Table** view.

Results in Table view for the Hacker News node



Results in Table view for the Hacker News node

Node executions

If a node executes successfully, a small green checkmark appears on top of the node in the canvas

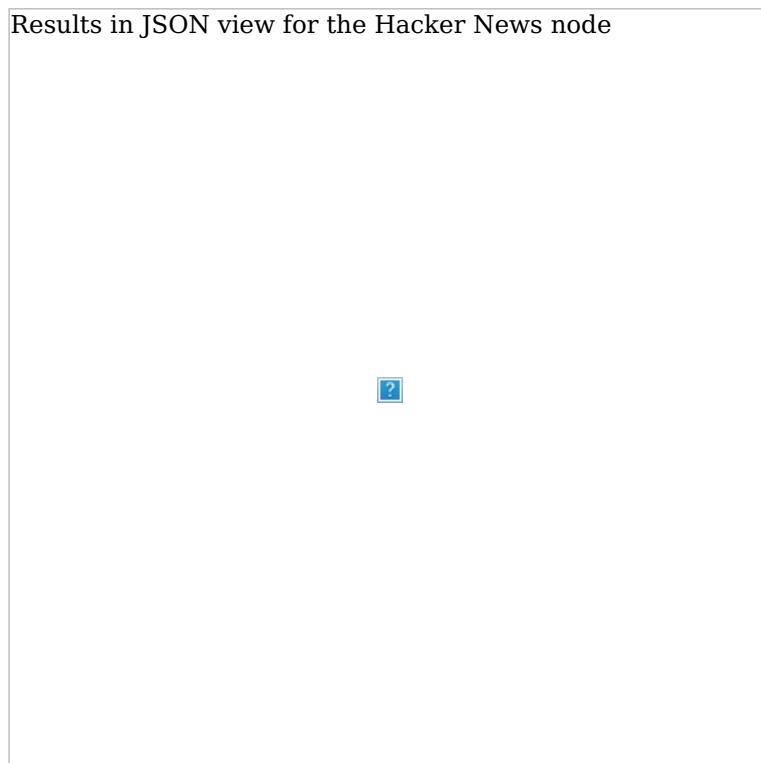
Successfully executed workflow



Successfully executed workflow

If there are no problems with the parameters and everything works fine, the requested data displays in the node window in **Table**, **JSON**, and **Schema** format. You can switch between these views by selecting the one you want from the **Table | JSON | Schema** button at the top of the node window.

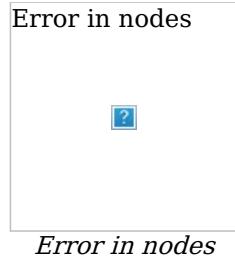
Here's our Hacker News output in JSON view:



Results in JSON view for the Hacker News node

The node window displays more information about the node execution:

- Next to the **Output** title, notice a small icon (this will be a green checkmark if the node execution succeeded). Beside it, there is an info icon. If you hover on it, you'll get two more pieces of information that can provide insights into the performance of each individual node in a workflow:
 - **Start Time:** When the node execution started.
 - **Execution Time:** How long it took for the node to return the results from the moment it started executing.
- Just below the **Output** title, you'll notice another piece of information: **10 items**. This field displays the number of items (records) that the node request returned. In this example, it's expected to be 10, since this is the limit we set in step 2. But if you don't set a limit, it's useful to see how many records are actually returned.



5. Save the workflow

Once you're finished editing the node, select **Back to canvas** to return to the main canvas.

By default, your workflow is automatically saved as "My workflow."

For this lesson, rename the workflow to be "Hacker News workflow."

Once you've renamed the workflow, be sure to save it.

There are two ways in which you can save a workflow:

- From the Canvas in Editor UI, click **Ctrl + S** or **Cmd + S** on your keyboard.
- Select the **Save** button in the top right corner of the Editor UI. You may need to leave the node editor first by clicking outside the dialog.

If you see a grey **Saved** text instead of the **Save** button, your workflow was automatically saved.

Summary

Congratulations, you just built your first workflow! In this lesson, you learned how to use actions in app nodes, configure their parameters and settings, and save and execute your workflow.

In the next lesson, you'll meet your new client, Nathan, who needs to automate his sales reporting work. You will build a more complex workflow for his use case, helping him become more productive at work.

Automating a (Real-world) Use Case

Meet Nathan 🎉. Nathan works as an Analytics Manager at ABCorp. His job is to support the ABCorp team with reporting and analytics. Being a true jack of all trades, he also handles several miscellaneous initiatives.

Some things that Nathan does are repetitive and mind-numbing. He wants to automate some of these tasks so that he doesn't burn out. As an **Automation Expert**, you are meeting with Nathan today to help him understand how he can offload some of his responsibilities to n8n.

Understanding the scenario

You ✨: Nice to meet you, Nathan. Glad to be doing this! What's a repetitive task that's error-prone and that you'd like to get off your plate first?

Nathan 🧑: Thanks for coming in! The most annoying one's gotta be the weekly sales reporting.

I have to collect sales data from our legacy data warehouse, which manages data from the main business processes of an organization, such as sales or production. Now, each sales order can have the status Processing or Booked. I have to calculate the sum of all the Booked orders and announce them in the company Discord every Monday. Then I have to create a spreadsheet of all the Processing sales so that the Sales Managers can review them and check if they need to follow up with customers.

This manual work is tough and requires high attention to detail to make sure that all the numbers are right. Inevitably, I lose my focus and mistype a number or I don't get it done on time. I've been criticized once by my manager for miscalculating the data.

You ✨: Oh no! Doesn't the data warehouse have a way to export the data?

Nathan 🧑: The data warehouse was written in-house ages ago. It doesn't have a CSV export but they recently added a couple of API endpoints that expose this data, if that helps.

You ✨: Perfect! That's a good start. If you have a generic API, we can add some custom code and a couple of services to make an automated workflow. This gig has n8n written all over it. Let's get started!

Designing the Workflow

Now that we know what Nathan wants to automate, let's consider the steps he needs to take to achieve his goals:

1. Get the relevant data (order id, order status, order value, employee name) from the data warehouse
2. Filter the orders by their status (Processing or Booked)
3. Calculate the total value of all the Booked orders
4. Notify the team members about the Booked orders in the company's Discord channel
5. Insert the details about the Processing orders in Airtable for follow-up
6. Schedule this workflow to run every Monday morning

Nathan's workflow involves sending data from the company's data warehouse to two external services:

- Discord
- Airtable

Before that, the data has to be wrangled with general functions (conditional filtering, calculation, scheduling).

n8n provides integrations for all these steps, so Nathan's workflow in n8n would look like this:

You will build this workflow in eight steps:

1. [Getting data from the data warehouse](#)
2. [Inserting data into Airtable](#)
3. [Filtering orders](#)
4. [Setting values for processing orders](#)
5. [Calculating booked orders](#)
6. [Notifying the team](#)
7. [Scheduling the workflow](#)
8. [Publishing and examining the workflow](#)

To build this workflow, you will need the credentials found in the email you received from n8n when you signed up for this course. If you haven't signed up already, you can do it [here](#). If you haven't received a confirmation email after signing up, [contact us](#).

[Start building!](#)

1. Getting data from the data warehouse

In this part of the workflow, you will learn how to get data by making HTTP requests with the [HTTP Request](#) node.

After completing this section, your workflow will look like this:

First, let's set the scene for building Nathan's workflow.

Create new workflow

Open your Editor UI and create a new workflow with one of the two possible commands:

- Select ++ctrl+alt+n++ or ++cmd+option+n++ on your keyboard.
- Open the left menu, navigate to **Workflows**, and select **Add workflow**.

Name this new workflow "Nathan's workflow."

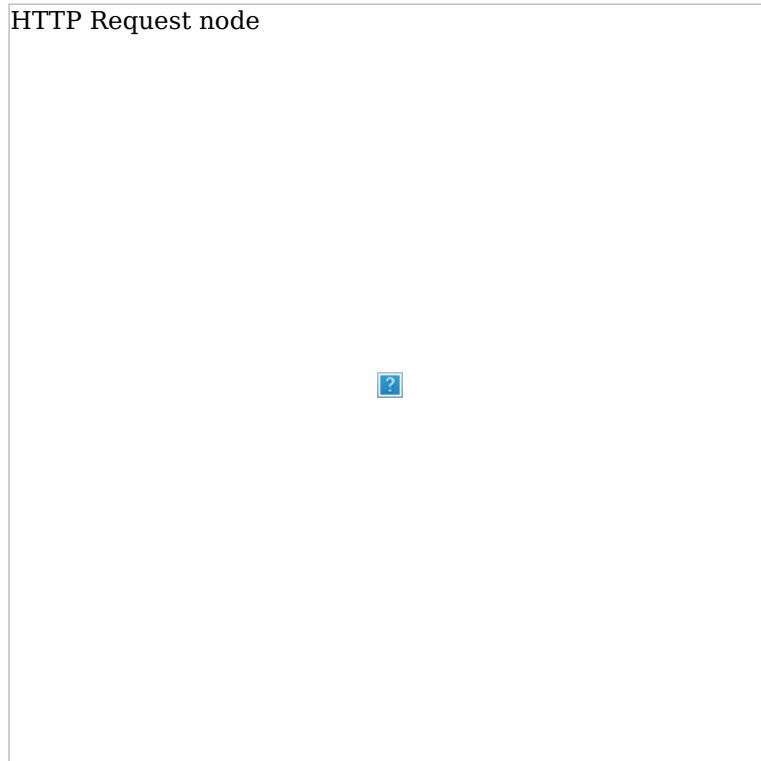
The first thing you need to do is get data from ABCorp's old data warehouse.

In a previous chapter, you used an action node designed for a specific service (Hacker News). But not all apps or services have dedicated nodes, like the legacy data warehouse from Nathan's company.

Though we can't directly export the data, Nathan told us that the data warehouse has a couple of API endpoints. That's all we need to access the data using the [HTTP Request](#) node in n8n.

Add an HTTP Request node

Now, in your Editor UI, add an HTTP Request node like you learned in the lesson [Adding nodes](#). The node window will open, where you need to configure some parameters.



HTTP Request node

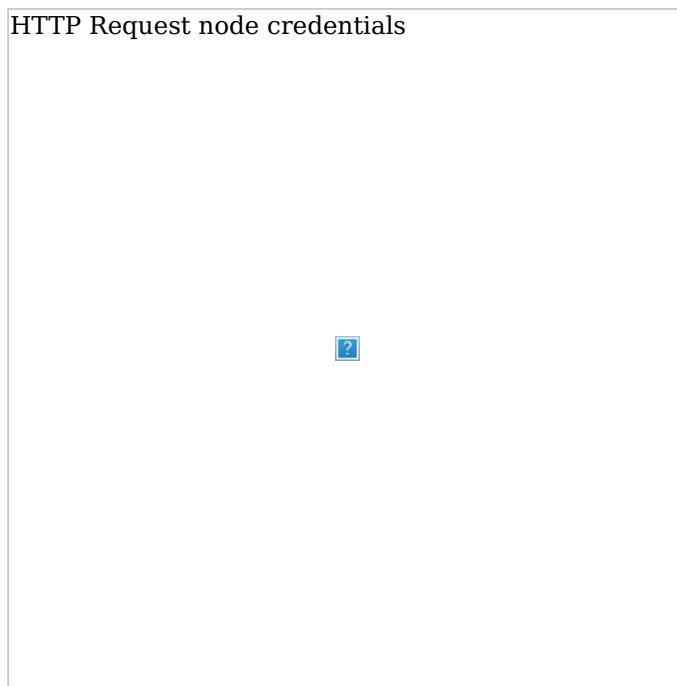
This node will use credentials.

In this case, you'll need the credentials for the ABCorp data warehouse API included in the email from n8n you received when you signed up for this course. If you haven't signed up yet, [sign up here](#).

In the **Parameters** of the HTTP Request node, make the following adjustments:

- **Method:** This should default to GET. Make sure it's set to GET.
- **URL:** Add the **Dataset URL** you received in the email when you signed up for this course.
- **Send Headers:** Toggle this control to true. In **Specify Headers**, ensure **Using Fields Below** is selected.
 - **Header Parameters > Name:** Enter `unique_id`.
 - **Header Parameters > Value:** The Unique ID you received in the email when you signed up for this course.
- **Authentication:** Select **Generic Credential Type**. This option requires credentials before allowing you to access the data.
 - **Generic Auth Type:** Select **Header Auth**. (This field will appear after you select the Generic Credential Type for the Authentication.)
 - **Credential for Header Auth:** To add your credentials, select **+ Create new credential**. This will open the Credentials window.
 - In the Credentials window, set **Name** to be the **Header Auth name** you received in the email when you signed up for this course.

- In the Credentials window, set **Value** to be the **Header Auth value** you received in the email when you signed up for this course.
- Select the **Save** button in the Credentials window to save your credentials. Your **Credentials Connection** window should look like this:



HTTP Request node credentials

Once you save, exit out of the Credentials window to return to the HTTP Request node.

Get the data

Select the **Execute step** button in the HTTP Request node window. The table view of the HTTP request results should look like this:

HTTP Request node output



HTTP Request node output

This view should be familiar to you from the [Building a mini-workflow](#) page.

This is the data from ABCorp's data warehouse that Nathan needs to work with. This data set includes sales information from 30 customers with five columns:

- **orderID**: The unique id of each order.
- **customerID**: The unique id of each customer.
- **employeeName**: The name of Nathan's colleague responsible for the customer.
- **orderPrice**: The total price of the customer's order.
- **orderStatus**: Whether the customer's order status is booked or still in processing.

What's next?

Nathan : This is great! You already automated an important part of my job with only one node. Now instead of manually accessing the data every time I need it, I can use the HTTP Request Node to automatically get the information.

You : Exactly! In the next step, I'll help you one step further and insert the data you retrieved into Airtable.

2. Inserting data into Airtable

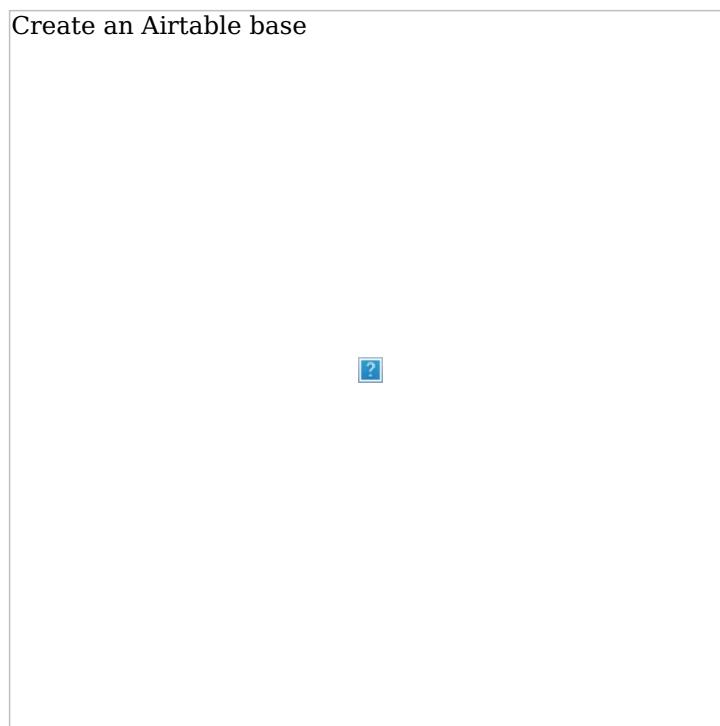
In this step of the workflow, you will learn how to insert the data received from the HTTP Request node into Airtable using the [Airtable node](#).

After this step, your workflow should look like this:

Configure your table

If we're going to insert data into Airtable, we first need to set up a table there. To do this:

1. [Create an Airtable account](#).
2. In your Airtable workspace add a new base from scratch and name it, for example, *beginner course*.

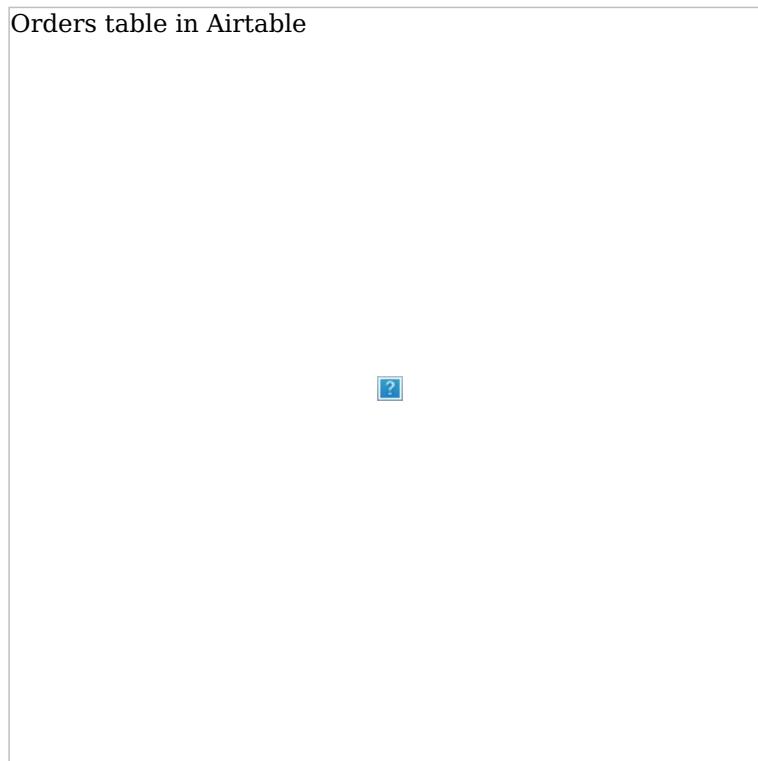


Create an Airtable base

3. In the beginner course base, by default, you have a table called **Table 1** with four fields: Name, Notes, Assignee, and Status. These fields aren't relevant for us since they aren't in our "orders" data set. This brings us to the next point: the names of the fields in Airtable have to match the names of the columns in the node result. Prepare the table by doing the following:
 - Rename the table from **Table 1** to **orders** to make it easier to identify.
 - Delete the 3 blank records created by default.
 - Delete the Notes, Assignee, and Status fields.
 - Edit the Name field (the primary field) to read `orderID`, with the **Number** field type.
 - Add the rest of the fields, and their field types, using the table below as a reference:

Field name	Field type
orderID	Number
customerID	Number
employeeName	Single line text
orderPrice	Number
orderStatus	Single line text

Now your table should look like this:



Orders table in Airtable

Now that the table is ready, let's return to the workflow in the n8n Editor UI.

Add an Airtable node to the HTTP Request node

Add an Airtable node connected to the HTTP Request node.

In the node panel:

1. Search for Airtable.
2. Select **Create a record** from the **Record Actions** search results.

This will add the Airtable node to your canvas and open the node details window.

In the Airtable node window, configure the following parameters:

- **Credential to connect with:**
 - Select **Create new credential**.

- Keep the default option **Connect using: Access Token** selected.
- **Access token:** Follow the instructions from the [Airtable credential](#) page to create your token. Use the recommended scopes and add access to your beginners course base. Save the credential and close the Credential window when you're finished.
- **Resource:** Record.
- **Operation:** Create. This operation will create new records in the table.
- **Base:** You can pick your base from a list (for example, beginner course).
- **Table:** orders.
- **Mapping Column Mode:** Map automatically. In this mode, the incoming data fields must have the same as the columns in Airtable.

Test the Airtable node

Once you've finished configuring the Airtable node, execute it by selecting **Execute step**. This might take a moment to process, but you can follow the progress by viewing the base in Airtable.

Your results should look like this:



Airtable node results

All 30 data records will now appear in the orders table in Airtable:

Imported records in the orders table



Imported records in the orders table

What's next?

Nathan 🤖: Wow, this automation is already so useful! But this inserts all collected data from the HTTP Request node into Airtable. Remember that I actually need to insert only processing orders in the table and calculate the price of booked orders?

You 🚗: Sure, no problem. As a next step, I'll use a new node to filter the orders based on their status.

3. Filtering Orders

In this step of the workflow, you will learn how to filter data using conditional logic and how to use expressions in nodes using the If node.

After this step, your workflow should look like this:

To insert only processing orders into Airtable we need to filter our data by `orderStatus`. Basically, we want to tell the program that *if* the `orderStatus` is *processing*, *then* insert all records with this status into Airtable; *else*, for example, if the `orderStatus` isn't *processing*, calculate the sum of all orders with the other `orderStatus` (*booked*).

This if-then-else command is conditional logic. In n8n workflows, you can add conditional logic with the If node, which splits a workflow conditionally based on comparison operations.

Add If node before the Airtable node

First, let's add an If node between the connection from the HTTP Request node to the Airtable node:

1. Hover over the arrow connection the **HTTP Request** node and the **Airtable** node.
2. Select the **+** sign between the HTTP Request node and the Airtable node.

Configure the If node

Selecting the plus removes the connection to the Airtable node to the HTTP request. Now, let's add an If node connected to the HTTP Request node:

1. Search for the If node.
2. Select it when it appears in the search.

For the If node, we'll use an expression.

In the If node window, configure the parameters:

- Set the value1 placeholder to `{{ $json.orderStatus } }` with the following steps:
 1. Hover over the value1 field.
 2. Select the **Expression** tab on the right side of the value1 field.
 3. Next, open the expression editor by selecting the link icon:

Opening the Expression Editor

[2]

Opening the Expression Editor

4. Use the left-side panel to select **HTTP Request > orderStatus** and drag it into the **Expression** field in the center of the window.

Expression Editor in the IF node



Expression Editor in the If node

5. Once you add the expression, close the **Edit Expression** dialog.
 - **Operation:** Select **String > is equal to**
 - Set the value2 placeholder to processing.

Select **Execute step** to test the If node.

Your results should look like this:

If node output



If node output

Note that the orders with a processing order status should show for the **True Branch** output, while the orders with a booked order status should show in the **False Branch** output.

Close the If node detail view when you're finished.

Insert data into Airtable

Next, we want to insert this data into Airtable. Remember what Nathan said at the end of the [Inserting data into Airtable](#) lesson?

I actually need to insert only processing orders in the table...

Since Nathan only needs the processing orders in the table, we'll connect the Airtable node to the If node's true connector.

In this case, since the Airtable node is already on our canvas, select the **If node** true connector and drag it to the Airtable node.

It's a good idea at this point to retest the Airtable node. Before you do, open your table in Airtable and delete all existing rows. Then open the Airtable node window in n8n and select **Execute step**.

Review your data in Airtable to be sure your workflow only added the correct orders (those with `orderStatus` of `processing`). There should be 14 records now instead of 30.

At this stage, your workflow should look like this:

What's next?

Nathan : This If node is so useful for filtering data! Now I have all the information about processing orders. I actually only need the `employeeName` and `orderID`, but I guess I can keep all the other fields just in case.

You : Actually, I wouldn't recommend doing that. Inserting more data requires more computational power, the data transfer is slower and takes longer, and takes up more storage resources in your table. In this particular case, 14 records with 5 fields might not seem like it'd make a significant difference, but if your business grows to thousands of records and dozens of fields, things add up and even one extra column can affect performance.

Nathan : Oh, that's good to know. Can you select only two fields from the processing orders?

You : Sure, I'll do that in the next step.

4. Setting Values for Processing Orders

In this step of the workflow, you will learn how to select and set data before transferring it to Airtable using the Edit Fields (Set) node. After this step, your workflow should look like this:

The next step in Nathan's workflow is to filter the data to only insert the employeeName and orderID of all processing orders into Airtable.

For this, you need to use the [Edit Fields \(Set\) node](#), which allows you to select and set the data you want to transfer from one node to another.

Add another node before the Airtable node

In your workflow, add another node before the **Airtable node** from the **If node** in the same way we did it in the [Filtering Orders](#) lesson on the If node's true connector. Feel free to drag the Airtable node further away if your canvas feels crowded.

Configure the Edit Fields node

Now search for the **Edit Fields (Set) node** after you've selected the + sign coming off the If node's true connector.

With the Edit Fields node window open, configure these parameters:

- Ensure **Mode** is set to **Manual Mapping**.
- While you can use the **Expression editor** we used in the [Filtering Orders](#) lesson, this time, let's drag the fields from the **Input** into the **Fields to Set**:
 - Drag **If > orderID** as the first field.
 - Drag **If > employeeName** as the second field.
- Ensure that **Include Other Input Fields** is set to false.

Select **Execute step**. You should see the following results:

Edit Fields (Set) node



Edit Fields (Set) node

Add data to Airtable

Next, let's insert these values into Airtable:

1. Go to your Airtable base.
2. Add a new table called `processingOrders`.
3. Replace the existing columns with two new columns:
 - `orderID` (primary field): Number
 - `employeeName`: Single line text
4. Delete the three empty rows in the new table.
5. In n8n, connect the Edit Fields node** connector to the **Airtable node**.
6. Update the Airtable node configuration to point to the new `processingOrders` table instead of the `orders` table.
7. Test your Airtable node to be sure it inserts records into the new `processingOrders` table.

At this stage, your workflow should now look like this:

What's next?

Nathan : You've already automated half of my work! Now I still need to calculate the booked orders for my colleagues. Can we automate that as well?

You : Yes! In the next step, I'll use some JavaScript code in a node to calculate the booked orders.

5. Calculating Booked Orders

In this step of the workflow you will learn how n8n structures data and how to add custom JavaScript code to perform calculations using the Code node. After this step, your workflow should look like this:

The next step in Nathan's workflow is to calculate two values from the booked orders:

- The total number of booked orders
- The total value of all booked orders

To calculate data and add more functionality to your workflows you can use the Code node, which lets you write custom JavaScript code.

About the Code node

In n8n, the data that's passed between nodes is an array of objects with the following JSON structure:

```
[  
  {  
    "json": { // (1)  
      "apple": "beets",  
      "carrot": {  
        "dill": 1  
      }  
    },  
    "binary": { // (2)  
      "apple-picture": { // (3)  
        "data": "...", // (4)  
        "mimeType": "image/png", // (5)  
        "fileExtension": "png", // (6)  
        "fileName": "example.png", // (7)  
      }  
    },  
    ...  
  ]
```

1. (required) n8n stores the actual data within a nested json key. This property is required, but can be set to anything from an empty object (like {}) to arrays and deeply nested data. The code node automatically wraps the data in a json object and parent array ([] if it's missing).
2. (optional) Binary data of item. Most items in n8n don't contain binary data.
3. (required) Arbitrary key name for the binary data.
4. (required) Base64-encoded binary data.
5. (optional) Should set if possible.
6. (optional) Should set if possible.
7. (optional) Should set if possible.

You can learn more about the expected format on the [n8n data structure](#) page.

Configure the Code node

Now let's see how to accomplish Nathan's task using the Code node.

In your workflow, add a **Code node** connected to the false branch of the **If node**.

With the Code node window open, configure these parameters:

- **Mode:** Select **Run Once for All Items**.
- **Language:** Select **JavaScript**.
- Copy the Code below and paste it into the **Code** box to replace the existing code:

```
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} }];
```

Notice the format in which we return the results of the calculation:

```
return [{ json: {totalBooked, bookedSum} }]
```

Now select **Execute step** and you should see the following results:

Code node output

[?]

Code node output

What's next?

Nathan 🧑: Wow, the Code node is powerful! This means that if I have some basic JavaScript skills I can power up my workflows.

You 🚀: Yes! You can progress from no-code to low-code!

Nathan 🧑: Now, how do I send the calculations for the booked orders to my team's Discord channel?

You 🚀: There's an n8n node for that. I'll set it up in the next step.

6. Notifying the Team

In this step of the workflow, you will learn how to send messages to a Discord channel using the [Discord node](#). After this step, your workflow should look like this:

Now that you have a calculated summary of the booked orders, you need to notify Nathan's team in their Discord channel. For this workflow, you will send messages to the [n8n server](#) on Discord.

Before you begin the steps below, use the link above to connect to the n8n server on Discord. Be sure you can access the `#course-level-1` channel.

In your workflow, add a Discord node connected to the Code node.

When you search for the Discord node, look for **Message Actions** and select **Send a message** to add the node.

In the Discord node window, configure these parameters:

- **Connection Type:** Select **Webhook**.
- **Credential for Discord Webhook:** Select - **Create New Credential** -
 - Copy the **Webhook URL** from the email you received when you signed up for this course and paste it into the **Webhook URL** field of the credentials.
 - Select **Save** and then close the credentials dialog.
- **Operation:** Select **Send a Message**.
- **Message:**
 - Select the **Expression** tab on the right side of the Message field.
 - Copy the text below and paste it into the **Expression** window, or construct it manually using the **Expression Editor**. This week we've `>{{$json["totalBooked"]}}` booked orders with a total value of `>{{$json["bookedSum"]}}`. My Unique ID: `{{$('HTTP Request').params["headerParameters"]["parameters"][0]["value"]}}`

Now select **Execute step** in the Discord node. If all works well, you should see this output in n8n:

Discord node output



Discord node output

And your message should appear in the Discord channel #course-level-1:

Discord message



Discord message

What's next?

Nathan 🧑: Incredible, you've saved me hours of tedious work already! Now I can execute this workflow when I need it. I just need to remember to run it every Monday morning at 9 AM.

You ✨: Don't worry about that, you can actually schedule the workflow to run on a specific day, time, or interval. I'll set this up in the next step.

7. Scheduling the Workflow

In this step of the workflow, you will learn how to schedule your workflow so that it runs automatically at a set time/interval using the Schedule Trigger node. After this step, your workflow should look like this:

The workflow you've built so far executes only when you click on **Execute Workflow**. But Nathan needs it to run automatically every Monday morning. You can do this with the Schedule Trigger, which allows you to schedule workflows to run periodically at fixed dates, times, or intervals.

To achieve this, we'll remove the Manual Trigger node we started with and replace it with a Schedule Trigger node instead.

Remove the Manual Trigger node

First, let's remove the Manual Trigger node:

1. Select the Manual Trigger node connected to your HTTP Request node.
2. Select the trash can icon to delete.

This removes the Manual Trigger node and you'll see an "Add first step" option.

Add the Schedule Trigger node

1. Open the nodes panel and search for **Schedule Trigger**.
2. Select it when it appears in the search results.

In the Schedule Trigger node window, configure these parameters:

- **Trigger Interval:** Select **Weeks**.
- **Weeks Between Triggers:** Enter 1.
- **Trigger on weekdays:** Select **Monday** (and remove **Sunday** if added by default).
- **Trigger at Hour:** Select **9am**.
- **Trigger at Minute:** Enter 0.

Your Schedule Trigger node should look like this:

Schedule Trigger Node



Schedule Trigger Node

Connect the Schedule Trigger node

Return to the canvas and connect your Schedule Trigger node to the HTTP Request node by dragging the arrow from it to the HTTP Request node.

Your full workflow should look like this:

What's next?

You 🤷: That was it for the workflow! I've added and configured all necessary nodes. Now every time you click on **Execute workflow**, n8n will execute all the nodes: getting, filtering, calculating, and transferring the sales data.

Nathan 🤔: This is just what I needed! My workflow will run automatically every Monday morning, correct?

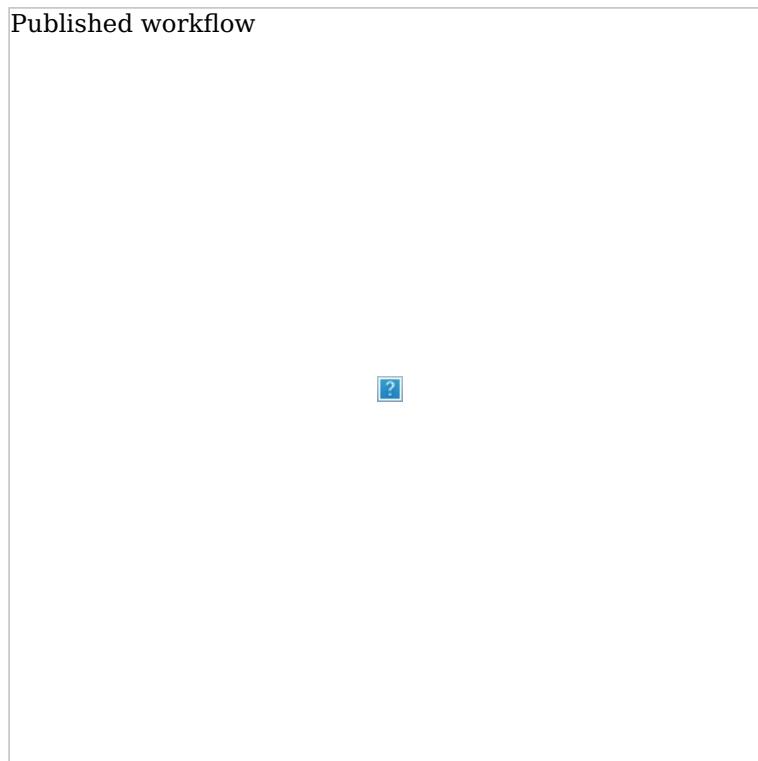
You 🤷: Not so fast. To do that, you need to publish your workflow. I'll do this in the next step and show you how to interpret the execution log.

8. Publishing and Examining the Workflow

In this step of the workflow, you will learn how to publish your workflow and change the default workflow settings.

Publishing a workflow means that it will run automatically every time a trigger node receives input or meets a condition. By default, newly created workflows are not published.

To publish your workflow, click **Publish** in the top navigation of the Editor UI. Nathan's workflow will now be executed automatically every Monday at 9 AM:



Published workflow

Workflow Executions

An execution represents a completed run of a workflow, from the first to the last node. n8n logs workflow executions, allowing you to see if the workflow succeeded or not. The execution log is useful for debugging your workflow and seeing at what stage it runs into issues.

To view the executions for a specific workflow, you can switch to the **Executions** tab when the workflow is open on the canvas. Use the **Editor** tab to swap back to the node editor.

To see the execution log for the entire n8n instance, in your Editor UI, select **Overview** and then select the **Executions** tab in the main panel.

Execution List



Execution List

The **Executions** window displays a table with the following information:

- **Name:** The name of the workflow
- **Started At:** The date and time when the workflow started
- **Status:** The status of the workflow (Waiting, Running, Succeeded, Cancelled, or Failed) and the amount of time it took the workflow to execute
- **Execution ID:** The ID of this workflow execution

Workflow Settings

You can customize your workflows and executions, or overwrite some global default settings in **Workflow Settings**.

Access these settings by selecting the three dots in the upper right corner of the Editor UI when the workflow is open on the canvas, then select **Settings**.

Workflow Settings



Workflow Settings

In the **Workflow Settings** window you can configure the following settings:

- **Execution Order:** Choose the execution logic for multi-branch workflows. You should leave this set to v1 if you don't have workflows that rely on the legacy execution ordering.
- **Error Workflow:** A workflow to run if the execution of the current workflow fails.
- **This workflow can be called by:** Workflows allowed to call this workflow using the [Execute Sub-workflow node](#).
- **Timezone:** The timezone to use in the current workflow. If not set, the global timezone. In particular, this setting is important for the [Schedule Trigger node](#), as you want to make sure that the workflow gets executed at the right time.
- **Save failed production executions:** If n8n should save the Execution data of the workflow when it fails. Default is to save.
- **Save successful production executions:** If n8n should save the Execution data of the workflow when it succeeds. Default is to save.
- **Save manual executions:** If n8n should save executions started from the Editor UI. Default is to save.
- **Save execution progress:** If n8n should save the execution data of each node. If set to Save, you can resume the workflow from where it stopped in case of an error, though keep in mind that this might make the execution slower. Default is to not save.
- **Timeout Workflow:** Whether to cancel a workflow execution after a specific period of time. Default is to not timeout.

What's next?

You 🤖: That was it! Now you have a 7-node workflow that will run automatically every Monday morning. You don't have to worry about remembering to wrangle the data. Instead, you can start your week with more meaningful or exciting work.

Nathan 🧑: This workflow is incredibly helpful, thank you! Now, what's next for you?

You 🤖: I'd like to build more workflows, share them with others, and use some workflows built by other people.

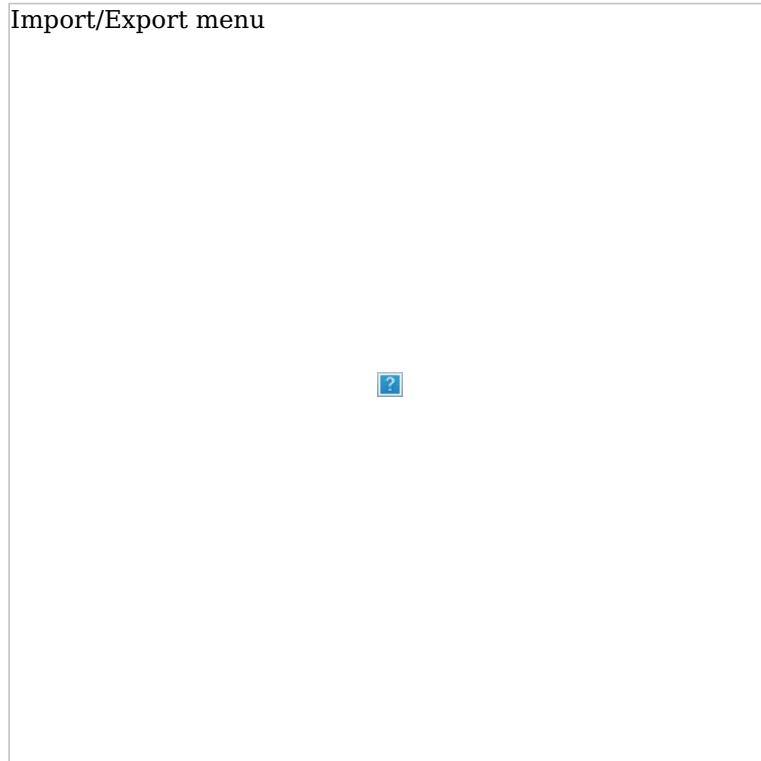
Exporting and importing workflows

In this chapter, you will learn how to export and import workflows.

Exporting and importing workflows

You can save n8n workflows locally as JSON files. This is useful if you want to share your workflow with someone else or import a workflow from someone else.

-8<- “_snippets/workflows/sharing-credentials.md”



Import & Export workflows menu

You can export and import workflows in three ways:

- From the **Editor UI** menu:
 - Export: From the top navigation bar, select the three dots in the upper right, then select **Download**. This will download your

- current workflow as a JSON file on your computer.
- Import: From the top navigation bar, select the three dots in the upper right, then select **Import from URL** (to import a published workflow) or **Import from File** (to import a workflow as a JSON file).
 - From the **Editor UI** canvas:
 - Export: Select all the nodes on the canvas and use `++ctrl+c++` to copy the workflow JSON. You can paste this into a file or share it directly with other people.
 - Import: You can paste a copied workflow JSON directly into the canvas with `++ctrl+v++`.
 - From the command line:
 - Export: See the [full list of commands](#) for exporting workflows or credentials.
 - Import: See the [full list of commands](#) for importing workflows or credentials.
-

Test your knowledge

Congratulations, you finished the n8n Course Level 1!

You've learned a lot about workflow automation and built your first business workflow. Why not showcase your skills?

You can test your knowledge by taking a **quiz**, which consists of questions about the theoretical concepts and workflows covered in this course.

- You need to have at least 80% correct answers in each part to pass the quiz.
- You can take the quiz as many times as you want.
- There's no time limit on answering the quiz questions.

[Take the quiz!](#)

What's next?

- Create new workflows for your work or personal use and share them with us. Don't have any ideas? Find inspiration on our [blog](#), [YouTube channel](#), [community forum](#), and [Discord server](#).
 - Take the n8n Course Level 2.
-

Level two: Introduction

Welcome to the **n8n Course Level 2!**

Is this course right for me?

This course is for you if you:

- Want to automate somewhat complex business processes.

- Want to dive deeper into n8n after taking the [Level 1 course](#).

What will I learn in this course?

The focus in this course is on working with data. You will learn how to:

- Use the data structure of n8n correctly.
- Process different data types (for example, XML, HTML, date, time, and binary data).
- Merge data from different sources (for example, a database, spreadsheet, or CRM).
- Use functions and JavaScript code in the [Code node](#).
- Deal with error workflows and workflow errors.

You will learn all this by completing short practical exercises after the theoretical explanations and building a business workflow following instructions.

What do I need to get started?

To follow along this course (at a comfortable pace) you will need the following:

- **n8n set up:** You can use the [self-hosted version](#) or [n8n Cloud](#).
- **A user ID:** [Sign up here](#) to get your unique ID and other credentials you will need in this course (Level 2). If you're a Level 1 finisher, please sign up again as you'll get different credentials for the Level 2 workflows.
- **Basic n8n skills:** We strongly recommend taking the [Level 1 course](#) before this one.
- **Basic JavaScript understanding**

How long does the course take?

Completing the course should take around **two hours**. You don't have to complete it in one go; feel free to take breaks and resume whenever you are ready.

How do I complete the course?

There are two milestones in this course that test your knowledge of what you have learned in the lessons:

- Building the [main workflow](#)
- Passing the [quiz](#) at the end of the course

You can always **check your progress** throughout the course by entering your unique ID [here](#).

If you successfully complete the milestones above, you will get a **badge and an avatar** in your forum profile. You can then share your profile and course verification ID to showcase your n8n skills to others.

[Let's get started!](#)

Understanding the data structure

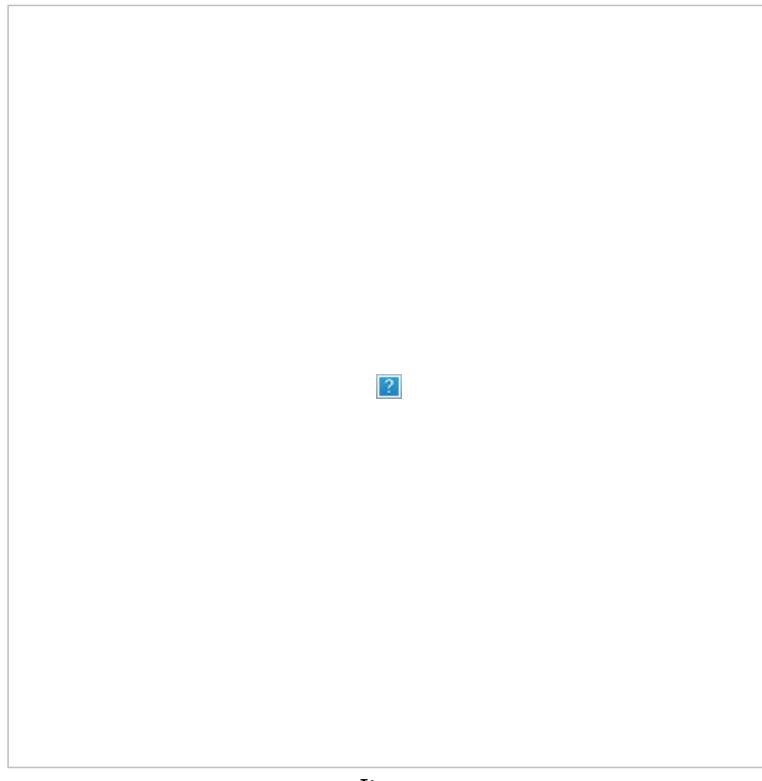
In this chapter, you will learn about the data structure of n8n and how to use the [Code node](#) to transform data and simulate node outputs.

Data structure of n8n

In a basic sense, n8n nodes function as an Extract, Transform, Load (ETL) tool. The nodes allow you to access (extract) data from multiple disparate sources, modify (transform) that data in a particular way, and pass (load) it along to where it needs to be.

The data that moves along from node to node in your workflow must be in a format (structure) that can be recognized and interpreted by each node. In n8n, this required structure is an array of objects.

Data sent from one node to another is sent as an array of JSON objects. The elements in this collection are called items.



Items

An n8n node performs its action on each item of incoming data.



Items in the Customer Datastore node

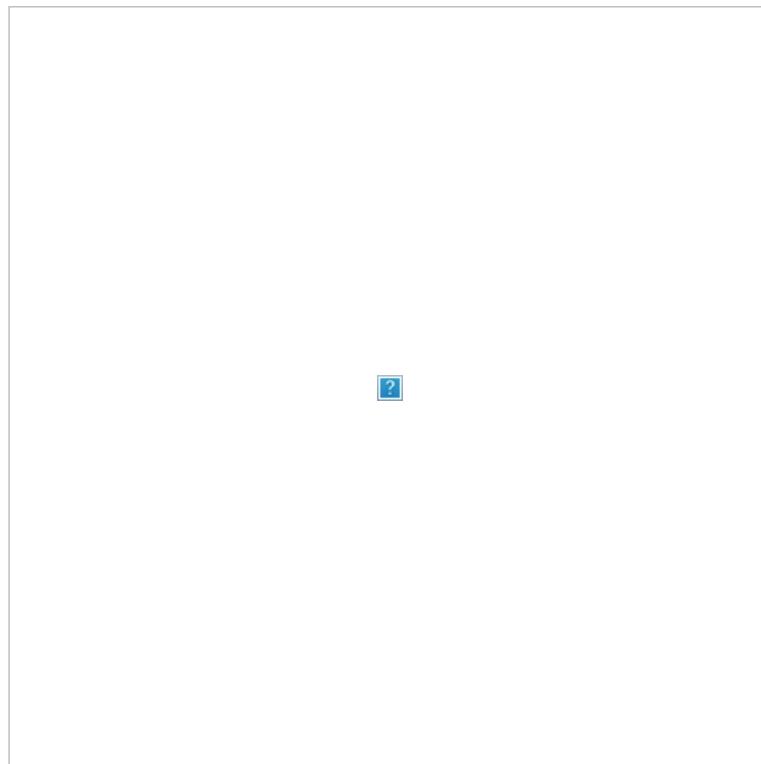
Creating data sets with the Code node

Now that you are familiar with the n8n data structure, you can use it to create your own data sets or simulate node outputs. To do this, use the [Code node](#) to write JavaScript code defining your array of objects

with the following structure:

```
return [
  {
    json: {
      apple: 'beets',
    }
  }
];
```

For example, the array of objects representing the Ninja turtles would look like this in the Code node:



Array of objects in the Code node

You can also have nested pairs, for example if you want to define a primary and a secondary color. In this case, you need to further wrap the key-value pairs in curly braces {}.

Exercise

In a Code node, create an array of objects named `myContacts` that contains the properties `name` and `email`, and the `email` property is further split into `personal` and `work`.

??? note “Show me the solution”

In the **Code node**, in the JavaScript Code field you have to write the following code:

```
```javascript
var myContacts = [
 {
 json: {
```

```

 name: 'Alice',
 email: {
 personal: 'alice@home.com',
 work: 'alice@wonderland.org'
 },
 },
},
{
 json: {
 name: 'Bob',
 email: {
 personal: 'bob@mail.com',
 work: 'contact@thebuilder.com'
 },
 }
},
];
```

```

When you execute the **Code node**, the result should look like this:

```
<figure><figcaption align = "center"><i>Result of Code node</i></figcaption></figure>
```

Referencing node data with the Code node

Just like you can use expressions to reference data from other nodes, you can also use some methods and variables in the **Code node**.

Please make sure you read these pages before continuing to the next exercise.

Exercise

Let's build on the previous exercise, in which you used the Code node to create a data set of two contacts with their names and emails. Now, connect a second Code node to the first one. In the new node, write code to create a new column named `workEmail` that references the work email of the first contact.

??? note “Show me the solution” In the **Code node**, in the JavaScript Code field you have to write the following code:

```
```javascript
let items = $input.all();
items[0].json.workEmail = items[0].json.email['work'];
return items;
```

```

When you execute the **Code node**, the result should look like this:

```
<figure>
<figcaption align = "center"><i>Code node reference</i></figcaption>
```

```
</figure>
```

Transforming data

The incoming data from some nodes may have a different data structure than the one used in n8n. In this case, you need to transform the data, so that each item can be processed individually.

The two most common operations for data transformation are:

- Creating multiple items from one item
- Creating a single item from multiple items

There are several ways to transform data for the purposes mentioned above:

- Use n8n's data transformation nodes. Use these nodes to modify the structure of incoming data that contain lists (arrays) without needing to use JavaScript code in the **Code node**:
 - Use the **Split Out node** to separate a single data item containing a list into multiple items.
 - Use the **Aggregate node** to take separate items, or portions of them, and group them together into individual items.
- Use the **Code node** to write JavaScript functions to modify the data structure of incoming data using the **Run Once for All Items** mode:
 - To create multiple items from a single item, you can use JavaScript code like this. This example assumes that the item has a key named `data` set to an array of items in the form of: [{ "data": [{<item_1>} , {<item_2>} , ...] }]:

```
return $input.first().json.data.map(item => {
    return {
        json: item
    }
});
```

- To create a single item from multiple items, you can use this JavaScript code:

```
return [
{
    json: {
        data_object: $input.all().map(item => item.json)
    }
}
];
```

These JavaScript examples assume your entire input is what you want to transform. As in the exercise above, you can also execute either operation on a specific field by identifying that in the items list, for example, if our `workEmail` example had multiple emails in a single field, we could run some code like this:

```
let items = $input.all();
return items[0].json.workEmail.map(item => {
    return {
        json: item
    }
});
```

Exercise

1. Use the **HTTP Request node** to make a GET request to the PokéAPI <https://pokeapi.co/api/v2/pokemon>. (This API requires no authentication).
2. Transform the data in the results field with the **Split Out node**.
3. Transform the data in the results field with the **Code node**.

??? note “Show me the solution”

1. To get the pokemon from the PokéAPI, execute the **HTTP Request node** with the following parameters:
 - **Authentication**: None
 - **Request Method**: GET
 - **URL**: <https://pokeapi.co/api/v2/pokemon>
2. To transform the data with the **Split Out node**, connect this node to the **HTTP Request node** and set the following parameters:
 - **Field To Split Out**: results
 - **Include**: No Other Fields
3. To transform the data with the **Code node**, connect this node to the **HTTP Request node** and write the following code in the JavaScript Code field:

```
```javascript
let items = $input.all();
return items[0].json.results.map(item => {
 return {
 json: item
 }
});```

```

---

## Processing different data types

In this chapter, you will learn how to process different types of data using [n8n core nodes](#).

### HTML and XML data

You’re most likely familiar with HTML and XML.

If you need to process HTML or XML data in your n8n workflows, use the [\*\*HTML node\*\*](#) or the [\*\*XML node\*\*](#).

Use the **HTML node** to extract HTML content of a webpage by referencing CSS selectors. This is useful if you want to collect structured information from a website (web-scraping).

### HTML Exercise

Let’s get the title of the latest n8n blog post:

1. Use the **HTTP Request node** to make a GET request to the URL <https://blog.n8n.io/> (this endpoint requires no authentication).
2. Connect an **HTML node** and configure it to extract the title of the first blog post on the page.
  - Hint: If you’re not familiar with CSS selectors or reading

HTML, the CSS selector `.post .item-title a` should help!

??? note "Show me the solution"

1. Configure the HTTP Request node with the following parameters:

- **Authentication**: None
- **Request Method**: GET
- **URL**: <https://blog.n8n.io/>

The result should look like this:

```
<figure><figcaption align = "center"><i>Result of
HTTP Request node</i></figcaption></figure>
```

2. Connect an **HTML node** to the **HTTP Request node** and configure the former's parameters:

- **Operation**: Extract HTML Content
- **Source Data**: JSON
- **JSON Property**: data
- **Extraction Values**:
  - **Key**: title
  - **CSS Selector**: ` .post .item-title a`
  - **Return Value**: HTML

You can add more values to extract more data.

The result should look like this:

```
<figure><figcaption align = "center"><i>Result of
HTML Extract node</i></figcaption></figure>
```

Use the **XML node** to convert XML to JSON and JSON to XML. This operation is useful if you work with different web services that use either XML or JSON and need to get and submit data between them in the two formats.

## XML Exercise

In the [final exercise of Chapter 1](#), you used an **HTTP Request node** to make a request to the PokéAPI. In this exercise, we'll return to that same API but we'll convert the output to XML:

1. Add an **HTTP Request node** that makes the same request to the PokéAPI at <https://pokeapi.co/api/v2/pokemon>.
2. Use the XML node to convert the JSON output to XML.

??? note "Show me the solution"

1. To get the pokemon from the PokéAPI, execute the **HTTP Request node** with the following parameters:

- **Authentication**: None
- **Request Method**: GET
- **URL**: <https://pokeapi.co/api/v2/pokemon>

2. Connect an **XML node** to it with the following parameters:

- **Mode**: JSON to XML
- **Property name**: data

The result should look like this:

```
<figure><figcaption align = "center">
<i>XML node (JSON to XML) – Table View</i></figcaption></figure>
```

To transform data the other way around, select the mode \*\*XML to JSON\*\*.

## Date, time, and interval data

Date and time data types include DATE, TIME, DATETIME, TIMESTAMP, and YEAR. The dates and times can be passed in different formats, for example:

- DATE: March 29 2022, 29-03-2022, 2022/03/29
- TIME: 08:30:00, 8:30, 20:30
- DATETIME: 2022/03/29 08:30:00
- TIMESTAMP: 1616108400 (Unix timestamp), 1616108400000 (Unix ms timestamp)
- YEAR: 2022, 22

There are a few ways you can work with dates and times:

- Use the **Date & Time node** to convert date and time data to different formats and calculate dates.
- Use **Schedule Trigger node** to schedule workflows to run at a specific time, interval, or duration.

Sometimes, you might need to pause the workflow execution. This might be necessary if you know that a service doesn't process the data instantly or it's slow to return all the results. In these cases, you don't want n8n to pass incomplete data to the next node.

If you run into situations like this, use the **Wait node** after the node that you want to delay. The **Wait node** pauses the workflow execution and will resume execution:

- At a specific time.
- After a specified time interval.
- On a webhook call.

## Date Exercise

Build a workflow that adds five days to an input date from the Customer Datastore node that you used before. Then, if the calculated date occurred after 1959, the workflow waits 1 minute before setting the calculated date as a value. The workflow should be triggered every 30 minutes.

To begin:

1. Add the **Customer Datastore (n8n training) node** with the **Get All People** action selected. Return All.
2. Add the **Date & Time node** to Round Up the created Date from the datastore to End of Month. Output this to field new-date. Include all input fields.
3. Add the **If node** to check if that new rounded date is after 1960-01-01 00:00:00.

4. Add the **Wait node** to the True output of that node and set it to wait for one minute.
5. Add the **Edit Fields (Set) node** to set a new field called `outputValue` to a String containing new-date. Include all input fields.
6. Add the **Schedule Trigger node** at the beginning of the workflow to trigger it every 30 minutes. (You can keep the [Manual Trigger node](#) for testing!)

??? note “Show me the solution”

1. Add the **Customer Datastore (n8n training) node** with the **Get All People** action selected.
  - Select the option to **Return All**.
2. Add a **Date & Time node** connected to the Customer Datastore node. Select the option to **Round a Date**.
  - Add the `created` date as the **Date** to round.
  - Select `Round Up` as the **Mode** and `End of Month` as the **To**.
  - Set the **Output Field Name** as `new-date`.
  - In **Options**, select **Add Option** and use the control to **Include Input Fields**.
3. Add an **If node** connected to the **Date & Time node**.
  - Add the new-date field as the first part of the condition.
  - Set the comparison to **Date & Time > is after**.
  - Add `1960-01-01 00:00:00` as the second part of the expression. (This should produce 3 items in the True Branch and 2 items in the False Branch)
4. Add a **Wait node** to the True output of the **If node**.
  - Set **Resume** to `After Time interval`.
  - Set **Wait Amount** to `1.00`.
  - Set **Wait Unit** to `Minutes`.
5. Add an **Edit Fields (Set) node** to the **Wait node**.
  - Use either JSON or Manual Mapping **Mode**.
  - Set a new field called `outputValue` to be the value of the new-date field.
  - Select the option to **Include Other Input Fields** and include **All** fields.
6. Add a **Schedule Trigger node** at the beginning of the workflow.
  - Set the **Trigger Interval** to use `Minutes`.
  - Set the **Minutes Between Triggers** to 30.
  - To test your schedule, be sure to publish the workflow.
  - Be sure to connect this node to the **Customer Datastore (n8n training) node** you began with!

The workflow should look like this:

```
<figure><figcaption align = "center"><i>Workflow for
transforming dates</i></figcaption></figure>
```

To check the configuration of each node, you can copy the JSON code of this workflow and either paste it into the Editor UI or save it as a file and import from file into a new workflow. See [Export and import workflows](/workflows/export-import.md) for more information.

```
```json
{
  "name": "Course 2, Ch 2, Date exercise",
  "nodes": [
```

```
{
  "parameters": {},
  "id": "6bf64d5c-4b00-43cf-8439-3cbf5e5f203b",
  "name": "When clicking \"Execute workflow\"",
  "type": "n8n-nodes-base.manualTrigger",
  "typeVersion": 1,
  "position": [
    620,
    280
  ],
  {
    "parameters": {
      "operation": "getAllPeople",
      "returnAll": true
    },
    "id": "a08a8157-99ee-4d50-8fe4-b6d7e16e858e",
    "name": "Customer Datastore (n8n training)",
    "type": "n8n-nodes-base.n8nTrainingCustomerDatastore",
    "typeVersion": 1,
    "position": [
      840,
      360
    ],
    {
      "parameters": {
        "operation": "roundDate",
        "date": "{ json.created }",
        "mode": "roundUp",
        "outputFieldName": "new-date",
        "options": {
          "includeInputFields": true
        }
      },
      "id": "f66a4356-2584-44b6-a4e9-1e3b5de53e71",
      "name": "Date & Time",
      "type": "n8n-nodes-base.dateTime",
      "typeVersion": 2,
      "position": [
        1080,
        360
      ],
      {
        "parameters": {
          "conditions": {
            "options": {
              "caseSensitive": true,
              "leftValue": "",
              "typeValidation": "strict"
            },
            "conditions": [
              {
                "id": "7c82823a-e603-4166-8866-493f643ba354",
                "leftValue": "{ json['new-date'] }",
                "rightValue": "1960-01-01T00:00:00",
                "operator": {
                  "type": "dateTime",
                  "operation": "after"
                }
              }
            ]
          }
        }
      }
    }
  }
}
```

```

        }
    }
],
"combinator": "and"
},
"options": {}
},
"id": "cea39877-6183-4ea0-9400-e80523636912",
"name": "If",
"type": "n8n-nodes-base.if",
"typeVersion": 2,
"position": [
    1280,
    360
]
},
{
"parameters": {
    "amount": 1,
    "unit": "minutes"
},
"id": "5aa860b7-c73c-4df0-ad63-215850166f13",
"name": "Wait",
"type": "n8n-nodes-base.wait",
"typeVersion": 1.1,
"position": [
    1480,
    260
],
"webhookId": "be78732e-787d-463e-9210-2c7e8239761e"
},
{
"parameters": {
    "assignments": {
        "assignments": [
            {
                "id": "e058832a-2461-4c6d-b584-043ecc036427",
                "name": "outputValue",
                "value": "= {{ $json['new-date'] }}",
                "type": "string"
            }
        ]
    },
    "includeOtherFields": true,
    "options": {}
},
"id": "be034e9e-3cf1-4264-9d15-b6760ce28f91",
"name": "Edit Fields",
"type": "n8n-nodes-base.set",
"typeVersion": 3.3,
"position": [
    1700,
    260
]
},
{
"parameters": {
    "rule": {
        "interval": [
            {

```

```
        "field": "minutes",
        "minutesInterval": 30
    }
]
}
},
"id": "6e8e4308-d0e0-4d0d-bc29-5131b57cf061",
"name": "Schedule Trigger",
"type": "n8n-nodes-base.scheduleTrigger",
"typeVersion": 1.1,
"position": [
    620,
    480
]
},
"pinData": {},
"connections": {
    "When clicking \"Execute workflow\": {
        "main": [
            [
                [
                    {
                        "node": "Customer Datastore (n8n training)",
                        "type": "main",
                        "index": 0
                    }
                ]
            ]
        ],
        "Customer Datastore (n8n training)": {
            "main": [
                [
                    [
                        {
                            "node": "Date & Time",
                            "type": "main",
                            "index": 0
                        }
                    ]
                ]
            ],
            "Date & Time": {
                "main": [
                    [
                        [
                            {
                                "node": "If",
                                "type": "main",
                                "index": 0
                            }
                        ]
                    ]
                ],
                "If": {
                    "main": [
                        [
                            [
                                {
                                    "node": "Wait",
                                    "type": "main",
                                    "index": 0
                                }
                            ]
                        ]
                    ]
                }
            }
        }
    }
}
```

```

    ]
},
"Wait": {
"main": [
[
{
    "node": "Edit Fields",
    "type": "main",
    "index": 0
}
]
],
"Schedule Trigger": {
"main": [
[
{
    "node": "Customer Datastore (n8n training)",
    "type": "main",
    "index": 0
}
]
]
}
}
```

```

## Binary data

Up to now, you have mainly worked with text data. But what if you want to process data that's not text, like images or PDF files? These types of files are represented in the binary numeral system, so they're considered binary data. In this form, binary data doesn't offer you useful information, so you'll need to convert it into a readable form.

In n8n, you can process binary data with the following nodes:

- [HTTP Request](#) to request and send files from/to web resources and APIs.
- [Read/Write Files from Disk](#) to read and write files from/to the machine where n8n is running.
- [Convert to File](#) to take input data and output it as a file.
- [Extract From File](#) to get data from a binary format and convert it to JSON.

To read or write a binary file, you need to write the path (location) of the file in the node's `File(s)` Selector parameter (for the Read operation) or in the node's `File Path` and `Name` parameter (for the Write operation).

## Binary Exercise 1

For our first binary exercise, let's convert a PDF file to JSON:

1. Make an HTTP request to get this PDF file:  
[https://media.kaspersky.com/pdf/Kaspersky\\_Lab\\_Whitepaper\\_Anti\\_blocker.pdf](https://media.kaspersky.com/pdf/Kaspersky_Lab_Whitepaper_Anti_blocker.pdf).
2. Use the **Extract From File node** to convert the file from binary to

JSON.

??? note "Show me the solution"

In the \*\*HTTP Request node\*\*, you should see the PDF file, like this:

```
<figure><figcaption align = "center"><i>HTTP
Request node to get PDF</i></figcaption></figure>
```

When you convert the PDF from binary to JSON using the \*\*Extract From File node\*\*, the result should look like this:

```
<figure><figcaption align = "center"><i>Extract
From File node</i></figcaption></figure>
```

To check the configuration of the nodes, you can copy the JSON workflow code below and paste it into your Editor UI:

```
```json
{
    "name": "Binary to JSON",
    "nodes": [
        {
            "parameters": {},
            "id": "78639a25-b69a-4b9c-84e0-69e045bed1a3",
            "name": "When clicking \"Execute Workflow\"",
            "type": "n8n-nodes-base.manualTrigger",
            "typeVersion": 1,
            "position": [
                480,
                520
            ],
            {
                "parameters": {
                    "url":
                    "https://media.kaspersky.com/pdf/Kaspersky_Lab_Whitepaper_Anti_blocker.pdf"
                },
                "options": {}
            },
            "id": "a11310df-1287-4e9a-b993-baa6bd4265a6",
            "name": "HTTP Request",
            "type": "n8n-nodes-base.httpRequest",
            "typeVersion": 4.1,
            "position": [
                700,
                520
            ],
            {
                "parameters": {
                    "operation": "pdf",
                    "options": {}
                },
                "id": "88697b6b-fb02-4c3d-a715-750d60413e9f",
                "name": "Extract From File",
                "type": "n8n-nodes-base.extractFromFile"
            }
        }
    ]
}
```

```

    "type": "n8n-nodes-base.extractFromFile",
    "typeVersion": 1,
    "position": [
        920,
        520
    ],
    "pinData": {},
    "connections": {
        "When clicking \"Execute Workflow\"": {
            "main": [
                [
                    [
                        {
                            "node": "HTTP Request",
                            "type": "main",
                            "index": 0
                        }
                    ]
                ]
            ],
            "HTTP Request": {
                "main": [
                    [
                        [
                            {
                                "node": "Extract From File",
                                "type": "main",
                                "index": 0
                            }
                        ]
                    ]
                ]
            }
        }
    }
```

```

## Binary Exercise 2

For our second binary exercise, let's convert some JSON data to binary:

1. Make an HTTP request to the Poetry DB API  
<https://poetrydb.org/random/1>.
2. Convert the returned data from JSON to binary using the **Convert to File node**.
3. Write the new binary file data to the machine where n8n is running using the **Read/Write Files From Disk node**.
4. To check that it worked out, use the **Read/Write Files From Disk node** to read the generated binary file.

??? note “Show me the solution”

The workflow for this exercise looks like this:

```

<figure><figcaption align = "center"><i>Workflow
for moving JSON to binary data</i></figcaption></figure>

```

To check the configuration of the nodes, you can copy the JSON workflow code below and paste it into your Editor UI:

```
```json
{
    "name": "JSON to file and Read-Write",
    "nodes": [
        {
            "parameters": {},
            "id": "78639a25-b69a-4b9c-84e0-69e045bed1a3",
            "name": "When clicking \"Execute Workflow\"",
            "type": "n8n-nodes-base.manualTrigger",
            "typeVersion": 1,
            "position": [
                480,
                520
            ],
            },
            {
            "parameters": {
                "url": "https://poetrydb.org/random/1",
                "options": {}
            },
            "id": "a11310df-1287-4e9a-b993-baa6bd4265a6",
            "name": "HTTP Request",
            "type": "n8n-nodes-base.httpRequest",
            "typeVersion": 4.1,
            "position": [
                680,
                520
            ],
            },
            {
            "parameters": {
                "operation": "toJson",
                "options": {}
            },
            "id": "06be18f6-f193-48e2-a8d9-35f4779d8324",
            "name": "Convert to File",
            "type": "n8n-nodes-base.convertToFile",
            "typeVersion": 1,
            "position": [
                880,
                520
            ],
            },
            {
            "parameters": {
                "operation": "write",
                "fileName": "/tmp/poetrydb.json",
                "options": {}
            },
            "id": "f2048e5d-fa8f-4708-b15a-d07de359f2e5",
            "name": "Read/Write Files from Disk",
            "type": "n8n-nodes-base.readWriteFile",
            "typeVersion": 1,
            "position": [
                1080,
                520
            ]
        ]
    ]
}
```

```
        },
        {
        "parameters": {
            "fileSelector": "={{ $json.fileName }}",
            "options": {}
        },
        "id": "d630906c-09d4-49f4-ba14-416c0f4de1c8",
        "name": "Read/Write Files from Disk1",
        "type": "n8n-nodes-base.readWriteFile",
        "typeVersion": 1,
        "position": [
            1280,
            520
        ]
    }
],
"pinData": {},
"connections": {
    "When clicking \"Execute Workflow\": {
        "main": [
            [
                {
                    "node": "HTTP Request",
                    "type": "main",
                    "index": 0
                }
            ]
        ],
        "HTTP Request": {
            "main": [
                [
                    {
                        "node": "Convert to File",
                        "type": "main",
                        "index": 0
                    }
                ]
            ]
        },
        "Convert to File": {
            "main": [
                [
                    {
                        "node": "Read/Write Files from Disk",
                        "type": "main",
                        "index": 0
                    }
                ]
            ]
        },
        "Read/Write Files from Disk": {
            "main": [
                [
                    {
                        "node": "Read/Write Files from Disk1",
                        "type": "main",
                        "index": 0
                    }
                ]
            ]
        }
    }
}
```

```
        ]  
        }  
    }  
}
```

Merging and splitting data

In this chapter, you will learn how to merge and split data, and in what cases it might be useful to perform these operations.

Merging data

In some cases, you might need to merge (combine) and process data from different sources.

Merging data can involve:

- Creating one data set from multiple sources.
- Synchronizing data between multiple systems. This could include removing duplicate data or updating data in one system when it changes in another.

In n8n, you can merge data from two different nodes using the [Merge node](#), which provides several merging options:

- [Append](#)
- [Combine](#)
 - [Merge by Fields](#): requires input fields to match on
 - [Merge by Position](#)
 - [Combine all possible combinations](#)
- [Choose Branch](#)

Notice that Combine > Merge by Fields requires you enter input fields to match on. These fields should contain identical values between the data sources so n8n can properly match data together. In the **Merge node**, they're called **Input 1 Field** and **Input 2 Field**.

Property Input fields in the Merge node



Property Input fields in the Merge node

Merge Exercise

Build a workflow that merges data from the Customer Datastore node and Code node.

1. Add a **Merge node** that takes Input 1 from a **Customer Datastore node** and Input 2 from a **Code node**.
2. In the **Customer Datastore node**, run the operation **Get All People**.
3. In the **Code node**, create an array of two objects with three properties: name, language, and country, where the property country has two sub-properties code and name.
 - Fill out the values of these properties with the information of two characters from the Customer Database.
 - For example, Jay Gatsby's language is English and country name is United States.
4. In the **Merge node**, try out different merge options.

??? note "Show me the solution"

The workflow for this exercise looks like this:

```
<figure><figcaption align = "center"><i>Workflow exercise for merging data</i></figcaption></figure>
```

If you merge data with the option ****Keep Matches**** using the name as the input fields to match, the result should look like this (note this example only contains Jay Gatsby; yours might look different depending on which characters you selected):

```
<figure><figcaption align = "center">
<i>Output of Merge node with option to keep matches</i></figcaption>
</figure>
```

To check the configuration of the nodes, you can copy the JSON workflow code below and paste it into your Editor UI:

```
```json
{
 "meta": {
 "templateCredsSetupCompleted": true,
 "instanceId": "cb484ba7b742928a2048bf8829668bed5b5ad9787579adea888f05980292a4a7"
 },
 "nodes": [
 {
 "parameters": {
 "mode": "combine",
 "mergeByFields": {
 "values": [
 {
 "field1": "name",
 "field2": "name"
 }
]
 },
 "options": {}
 },
 "id": "578365f3-26dd-4fa6-9858-f0a5fdfc413b",
 "name": "Merge",
 "type": "n8n-nodes-base.merge",
 "typeVersion": 2.1,
 "position": [
 720,
 580
]
 },
 {
 "parameters": {},
 "id": "71aa5aad-afdf-4f8a-bca0-34450eee8acc",
 "name": "When clicking \"Execute workflow\"",
 "type": "n8n-nodes-base.manualTrigger",
 "typeVersion": 1,
 "position": [
 260,
 560
]
 },
 {
 "parameters": {
 "operation": "getAllPeople"
 },
 "id": "497174fe-3cab-4160-8103-78b44efd038d",
 "name": "Customer Datastore (n8n training)",
 "type": "n8n-nodes-base.n8nTrainingCustomerDatastore",
 "typeVersion": 1,
 "position": [
 500,
 560
]
 }
]
}
```

```
 460
]
},
{
 "parameters": {
 "jsCode": "return [\n {\n 'name': 'Jay Gatsby',\n 'language': 'English',\n 'country': {\n 'code': 'US',\n 'name': 'United States'\n }\n }\n]\n },\n "id": "387e8a1e-e796-4f05-8e75-7ce25c786c5f",\n "name": "Code",\n "type": "n8n-nodes-base.code",\n "typeVersion": 2,\n "position": [
 500,
 720
]
},
],
"connections": {
 "When clicking \"Execute workflow\": {
 "main": [
 [
 {
 "node": "Customer Datastore (n8n training)",
 "type": "main",
 "index": 0
 },
 {
 "node": "Code",
 "type": "main",
 "index": 0
 }
]
],
 "Customer Datastore (n8n training)": {
 "main": [
 [
 [
 {
 "node": "Merge",
 "type": "main",
 "index": 0
 }
]
]
],
 "Code": {
 "main": [
 [
 [
 {
 "node": "Merge",
 "type": "main",
 "index": 1
 }
]
]
]
 }
 }
 },
 "pinData": {}
}
```

```
}
```

## Looping

In some cases, you might need to perform the same operation on each element of an array or each data item (for example sending a message to every contact in your address book). In technical terms, you need to iterate through the data (with loops).

n8n generally handles this repetitive processing automatically, as the nodes run once for each item, so you don't need to build loops into your workflows.

However, there are some exceptions of nodes and operations that will require you to build a loop into your workflow.

To create a loop in an n8n workflow, you need to connect the output of one node to the input of a previous node, and add an **If node** to check when to stop the loop.

## Splitting data in batches

If you need to process large volumes of incoming data, execute the **Code node** multiple times, or avoid API rate limits, it's best to split the data into batches (groups) and process these batches.

For these processes, use the **Loop Over Items node**. This node splits input data into a specified batch size and, with each iteration, returns a predefined amount of data.

### Loop/Batch Exercise

Build a workflow that reads the RSS feed from Medium and dev.to. The workflow should consist of three nodes:

1. A **Code node** that returns the URLs of the RSS feeds of Medium (<https://medium.com/feed/n8n-io>) and dev.to (<https://dev.to/feed/n8n>).
2. A **Loop Over Items node** with Batch Size: 1, that takes in the inputs from the **Code node** and **RSS Read node** and iterates over the items.
3. An **RSS Read node** that gets the URL of the Medium RSS feed, passed as an expression: {{ \$json.url }}.
  - The **RSS Read node** is one of the exception nodes which processes only the first item it receives, so the **Loop Over Items node** is necessary for iterating over multiple items.

??? note "Show me the solution"

1. Add a \*\*Code Node\*\*. You can format the code in several ways, one way is:
  - Set \*\*Mode\*\* to `Run Once for All Items`.
  - Set \*\*Language\*\* to `JavaScript`.
  - Copy the code below and paste it into the JavaScript Code editor:

```
```javascript
let urls = [
```

```

        {
            json: {
                url: 'https://medium.com/feed/n8n-io'
            }
        },
        {
            json: {
                url: 'https://dev.to/feed/n8n'
            }
        }
    ]
    return urls;
```
2. Add a **Loop Over Items node** connected to the **Code node**.
- Set **Batch Size** to `1`.
3. The **Loop Over Items node** automatically adds a node called "Replace Me". Replace that node with an **RSS Read node**.
- Set the **URL** to use the url from the Code Node: `{{ $json.url }}`.
```

The workflow for this exercise looks like this:

```
<figure><figcaption align = "center"><i>Workflow for getting RSS feeds from two blogs</i></figcaption></figure>
```

To check the configuration of the nodes, you can copy the JSON workflow code below and paste it into your Editor UI:

```
```json
{
  "meta": {
    "templateCredsSetupCompleted": true,
    "instanceId": "cb484ba7b742928a2048bf8829668bed5b5ad9787579adea888f05980292a4a7"
  },
  "nodes": [
    {
      "parameters": {},
      "id": "ed8dc090-ae8c-4db6-a93b-0fa873015c25",
      "name": "When clicking \"Execute workflow\"",
      "type": "n8n-nodes-base.manualTrigger",
      "typeVersion": 1,
      "position": [
        460,
        460
      ],
      {
        "parameters": {
          "jsCode": "let urls = [\n    {\n        json: {\n            url:\n                'https://medium.com/feed/n8n-io'\n        }\n    },\n    {\n        json: {\n            url:\n                'https://dev.to/feed/n8n'\n        }\n    }\n]\n\nreturn urls;"
        },
        "id": "1df2a9bf-f970-4e04-b906-92dbbc9e8d3a",
        "name": "Code",
        "type": "n8n-nodes-base.code",
        "typeVersion": 2,
      }
    }
  ]
}
```

```
"position": [
    680,
    460
],
{
"parameters": {
    "options": {}
},
"id": "3cce249a-0eab-42e2-90e3-dbdf3684e012",
"name": "Loop Over Items",
"type": "n8n-nodes-base.splitInBatches",
"typeVersion": 3,
"position": [
    900,
    460
]
},
{
"parameters": {
    "url": "={{ $json.url }}",
    "options": {}
},
"id": "50e1c1dc-9a5d-42d3-b7c0-accc31636aa6",
"name": "RSS Read",
"type": "n8n-nodes-base.rssFeedRead",
"typeVersion": 1,
"position": [
    1120,
    460
]
}
],
"connections": {
    "When clicking \"Execute workflow\": {
        "main": [
            [
                [
                    {
                        "node": "Code",
                        "type": "main",
                        "index": 0
                    }
                ]
            ]
        ],
        "Code": {
            "main": [
                [
                    [
                        {
                            "node": "Loop Over Items",
                            "type": "main",
                            "index": 0
                        }
                    ]
                ]
            ]
        },
        "Loop Over Items": {
            "main": [
                null,
                [

```

```
{
  "node": "RSS Read",
  "type": "main",
  "index": 0
}
]
},
"RSS Read": {
  "main": [
    [
      {
        "node": "Loop Over Items",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"pinData": {}
}
```

```

## Dealing with errors in workflows

Sometimes you build a nice workflow, but it fails when you try to execute it. Workflow executions may fail for a variety of reasons, ranging from straightforward problems with incorrectly configuring a node or a failure in a third-party service to more mysterious errors.

But don't panic. In this lesson, you'll learn how you can troubleshoot errors so you can get your workflow up and running as soon as possible.

## Checking failed workflows

n8n tracks executions of your workflows.

When one of your workflows fails, you can check the Executions log to see what went wrong. The Executions log shows you a list of the latest execution time, status, mode, and running time of your saved workflows.

Open the Executions log by selecting **Executions** in the left-side panel.

To investigate a specific failed execution from the list, select the name or the **View** button that appears when you hover over the row of the respective execution.

## Executions log

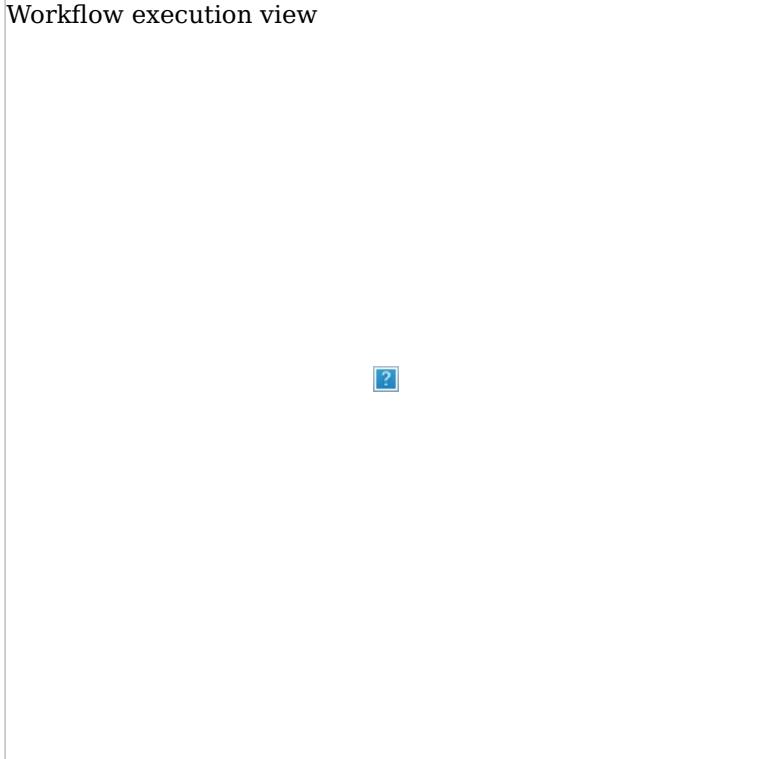


### *Executions log*

This will open the workflow in read-only mode, where you can see the execution of each node. This representation can help you identify at what point the workflow ran into issues.

To toggle between viewing the execution and the editor, select the **Editor | Executions** button at the top of the page.

Workflow execution view



*Workflow execution view*

## Catching erroring workflows

To catch failed workflows, create a separate **Error Workflow** with the **Error Trigger node**. This workflow will only execute if the main workflow execution fails.

Use additional nodes in your **Error Workflow** that make sense, like sending notifications about the failed workflow and its errors using email or Slack.

To receive error messages for a failed workflow, set the **Error Workflow** in the Workflow Settings to an Error Workflow that uses an **Error Trigger node**.

The only difference between a regular workflow and an Error Workflow is that the latter contains an **Error Trigger node**. Make sure to create this node before you set this as another workflow's designated Error Workflow.

### Exercise

In the previous chapters, you've built several small workflows. Now, pick one of them that you want to monitor and create an Error Workflow for it:

1. Create a new Error Workflow.
2. Add the **Error Trigger node**.
3. Connect a node for the communication platform of your choice to the Error Trigger node, like Slack, Discord, Telegram, or even Gmail or a more generic Send Email.

- In the workflow you want to monitor, open the [Workflow Settings](#) and select the new Error Workflow you just created. Note that this workflow needs to run automatically to trigger the error workflow.

??? note “Show me the solution”

The workflow for this exercise looks like this:

```
<figure><figcaption align = "center"><i>Error workflow</i></figcaption></figure>
```

To check the configuration of the nodes, you can copy the JSON workflow code below and paste it into your Editor UI:

```
```json
{
  "nodes": [
    {
      "parameters": {},
      "name": "Error Trigger",
      "type": "n8n-nodes-base.errorTrigger",
      "typeVersion": 1,
      "position": [
        720,
        -380
      ],
      {
        "parameters": {
          "channel": "channelname",
          "text": "=This workflow {{node[\"Error Trigger\"]}.json[\"workflow\"][\"name\"]} failed.\nHave a look at it here: {{$node[\"Error Trigger\"]}.json[\"execution\"][\"url\"]}}",
          "attachments": [],
          "otherOptions": {}
        },
        "name": "Slack",
        "type": "n8n-nodes-base.slack",
        "position": [
          900,
          -380
        ],
        "typeVersion": 1,
        "credentials": {
          "slackApi": {
            "id": "17",
            "name": "slack_credentials"
          }
        }
      }
    ],
    "connections": {
      "Error Trigger": {
        "main": [
          [
            {
              "node": "Slack",
              "type": "main",
              "index": 0
            }
          ]
        ]
      }
    }
  ],
  "connections": {
    "Error Trigger": {
      "main": [
        [
          {
            "node": "Slack",
            "type": "main",
            "index": 0
          }
        ]
      ]
    }
  }
}
```

```
        }
    ]
}
}...
```

Throwing exceptions in workflows

Another way of troubleshooting workflows is to include a **Stop and Error node** in your workflow. This node throws an error. You can specify the error type:

- **Error Message**: returns a custom message about the error
- **Error Object**: returns the type of error

You can only use the **Stop and Error node** as the last node in a workflow.

Automating a business workflow

Remember [our friend Nathan](#)?

Nathan 🧑: Hello, it's me again. My manager was so impressed with my first workflow automation solution that she entrusted me with more responsibility.

You 🚫: More work and responsibility. Congratulations, I guess. What do you need to do now?

Nathan 🧑: I got access to all our sales data and I'm now responsible for creating two reports: one for regional sales and one for orders prices. They're based on data from different sources and come in different formats.

You 🚫: Sounds like a lot of manual work, but the kind that can be automated. Let's do it!

Workflow design

Now that we know what Nathan wants to automate, let's list the steps he needs to take to achieve this:

1. Get and combine data from all necessary sources.
2. Sort the data and format the dates.
3. Write binary files.
4. Send notifications using email and Discord.

n8n provides [core nodes](#) for all these steps. This use case is somewhat complex. We should build it from three separate workflows:

1. A workflow that merges the company data with external information.
2. A workflow that generates the reports.
3. A workflow that monitors errors in the second workflow.

Workflow prerequisites

To build the workflows, you will need the following:

- An [Airtable](#) account and [credentials](#).
- A [Google](#) account and [credentials](#) to access Gmail.
- A [Discord](#) account and webhook URL (you receive this using email when you sign up for this course).

Next, you will build these three workflows with step-by-step instructions.

Workflow 1: Merging data

Nathan's company stores its customer data in Airtable. This data contains information about the customers' ID, country, email, and join date, but lacks data about their respective region and subregion. You need to fill in these last two fields in order to create the reports for regional sales.

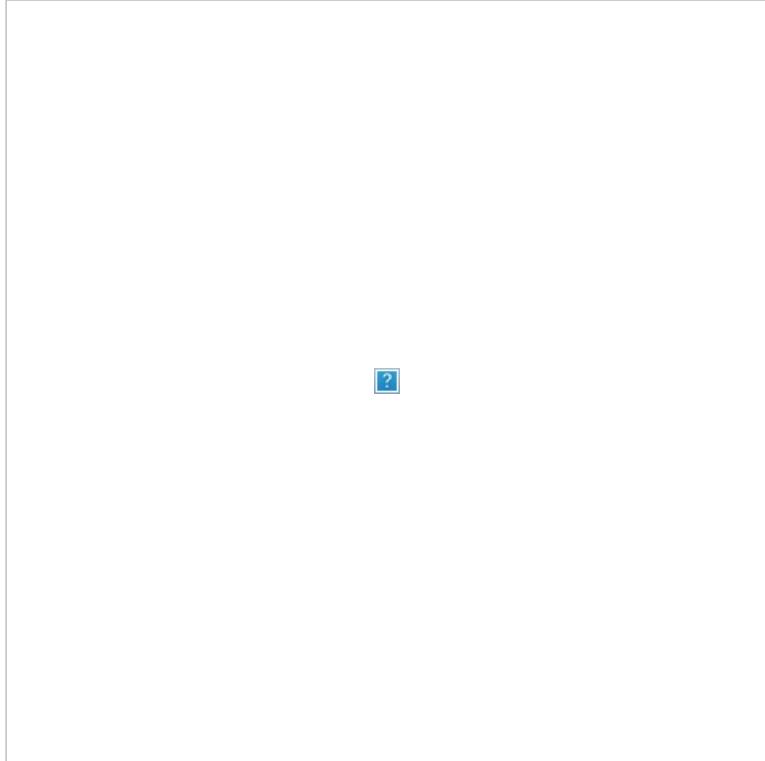
To accomplish this task, you first need to make a copy of this table in your Airtable account:

When setting up your Airtable, ensure that the `customerSince` column is configured as a Date type field with the **Include time** option enabled. Without this setting, you may encounter errors in step 4 when updating the table.

Next, build a small workflow that merges data from Airtable and a REST Countries API:

1. Use the **Airtable node** to list the data in the Airtable table named `customers`.
2. Use the **HTTP Request node** to get data from the REST Countries API: <https://restcountries.com/v3.1/all>, and send the query parameter `name` with the value `name,region,subregion`. This will return data about world countries, split out into separate items.
3. Use the **Merge node** to merge data from Airtable and the Countries API by country name, represented as `customerCountry` in Airtable and `name.common` in the Countries API, respectively.
4. Use another Airtable node to update the fields `region` and `subregion` in Airtable with the data from the Countries API.

The workflow should look like this:



Workflow 1 for merging data from Airtable and the Countries API

```
/// question | Quiz questions * How many items does the HTTP Request node return? * How many items does the Merge node return? * How many unique regions are assigned in the customers table? * What's the subregion assigned to the customerID 10? ///
```

Workflow 2: Generating reports

In this workflow, you will merge data from different sources, transform binary data, generate files, and send notifications about them. The final workflow should look like this:

Workflow 2 for aggregating data and generating files



Workflow 2 for aggregating data and generating files

To make things easier, let's split the workflow into three parts.

Part 1: Getting data from different sources

The first part of the workflow consists of five nodes:

Workflow 1: Getting data from different sources



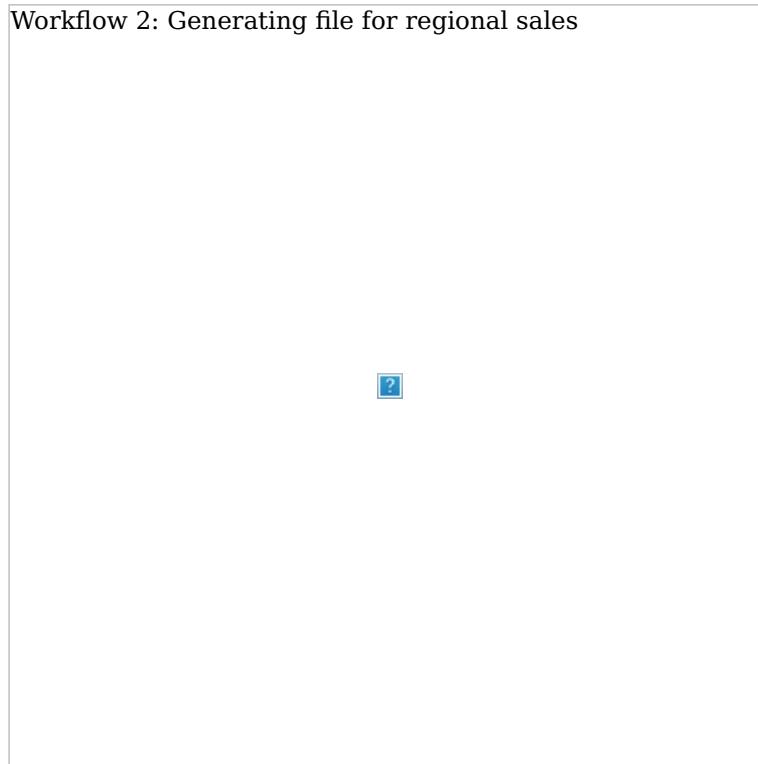
Workflow 1: Getting data from different sources

1. Use the **HTTP Request node** to get data from the API endpoint that stores company data. Configure the following node parameters:
 - **Method:** Get
 - **URL:** The **Dataset URL** you received in the email when you signed up for this course.
 - **Authentication:** Generic Credential Type
 - **Generic Auth Type:** Header Auth
 - **Credentials for Header Auth:** The Header Auth name and Header Auth value you received in the email when you signed up for this course.
 - **Send Headers:** Toggle to true
 - **Specify Headers:** Select Using Fields Below
 - **Name:** unique_id
 - **Value:** The unique ID you received in the email when you signed up for this course.
2. Use the **Airtable node** to list data from the customers table (where you updated the fields region and subregion).
3. Use the **Merge node** to merge data from the Airtable and HTTP Request node, based on matching the input fields for customerID.
4. Use the **Sort node** to sort data by orderPrice in descending order.

```
/// question | Quiz questions * What's the name of the employee assigned to customer 1? * What's the order status of customer 2? * What's the highest order price? ///
```

Part 2: Generating file for regional sales

The second part of the workflow consists of four nodes:



Workflow 2: Generating file for regional sales

1. Use the **If node** to filter to only display orders from the region Americas.
2. Use the **Convert to File** to transform the incoming data from JSON to binary format. Convert each item to a separate file. (Bonus points if you can figure out how to name each report based on the orderID!)
3. Use the **Gmail node** (or another email node) to send the files using email to an address you have access to. Note that you need to add an attachment with the data property.
4. Use the **Discord node** to send a message in the n8n Discord channel #course-level-two. In the node, configure the following parameters:
 - **Webhook URL:** The Discord URL you received in the email when you signed up for this course.
 - **Text:** "I sent the file using email with the label ID {label ID}. My ID:" followed by the unique ID emailed to you when you registered for this course.
Note that you need to replace the text in curly braces {} with expressions that reference the data from the nodes.

```
/// question | Quiz questions * How many orders are assigned to the Americas region? * What's the total price of the orders in the Americas region? * How many items does the Write Binary File node return?  
///
```

Part 3: Generating files for total sales

The third part of the workflow consists of five nodes:

Workflow 3: Generating files for total sales



Workflow 3: Generating files for total sales

1. Use the **Loop Over Items node** to split data from the Item Lists node into batches of 5.
2. Use the **Set node** to set four values, referenced with expressions from the previous node: customerEmail, customerRegion, customerSince, and orderPrice.
3. Use the **Date & Time node** to change the date format of the field customerSince to the format MM/DD/YYYY.
 - Set the **Include Input Fields** option to keep all the data together.
4. Use the **Convert to File node** to create a CSV spreadsheet with the file name set as the expression: `{{$runIndex > 0 ? 'file_low_orders':'file_high_orders'}}.`
5. Use the **Discord node** to send a message in the n8n Discord channel #course-level-two. In the node, configure the following parameters:
 - **Webhook URL:** The Discord URL you received in the email when you signed up for this course.
 - **Text:** "I created the spreadsheet {file name}. My ID:" followed by the unique ID emailed to you when you registered for this course.
Note that you need to replace {file name} with an expression that references data from the previous **Convert to File node**.

```
/// question | Quiz questions * What's the lowest order price in the first batch of items? * What's the formatted date of customer 7? * How many items does the Convert to File node return? ///
```

??? note “Show me the solution”

To check the configuration of the nodes, you can copy the JSON workflow code below and paste it into your Editor UI:

```

```json
{
 "meta": {
 "templateCredsSetupCompleted": true,
 "instanceId": "cb484ba7b742928a2048bf8829668bed5b5ad9787579adea888f05980292a4a7"
 },
 "nodes": [
 {
 "parameters": {
 "sendTo": "bart@n8n.io",
 "subject": "Your TPS Reports",
 "emailType": "text",
 "message": "Please find your TPS report attached.",
 "options": {
 "attachmentsUi": {
 "attachmentsBinary": [
 {}
]
 }
 }
 },
 "id": "d889eb42-8b34-4718-b961-38c8e7839ea6",
 "name": "Gmail",
 "type": "n8n-nodes-base.gmail",
 "typeVersion": 2.1,
 "position": [
 2100,
 500
],
 "credentials": {
 "gmailOAuth2": {
 "id": "HFeScCcFcn1NW81yu",
 "name": "Gmail account 7"
 }
 }
 },
 {
 "parameters": {},
 "id": "c0236456-40be-4f8f-a730-e56cb62b7b5c",
 "name": "When clicking \"Execute workflow\"",
 "type": "n8n-nodes-base.manualTrigger",
 "typeVersion": 1,
 "position": [
 780,
 600
]
 },
 {
 "parameters": {
 "url": "https://internal.users.n8n.cloud/webhook/level2-erp",
 "authentication": "genericCredentialType",
 "genericAuthType": "httpHeaderAuth",
 "sendHeaders": true,
 "headerParameters": {
 "parameters": [
 {
 "name": "unique_id",
 "value": "recFIcD6UlSyxaVMQ"
 }
]
 }
 }
 }
]
}
```

```

```

        }
    ]
},
"options": {}
},
"id": "cc106fa0-6630-4c84-aea4-a4c7a3c149e9",
"name": "HTTP Request",
"type": "n8n-nodes-base.httpRequest",
"typeVersion": 4.1,
"position": [
    1000,
    500
],
"credentials": {
    "httpHeaderAuth": {
        "id": "qeHdJdqqaTC69cm",
        "name": "Course L2 Credentials"
    }
},
{
"parameters": {
    "operation": "search",
    "base": {
        "__rl": true,
        "value": "apprtKkVasbQDbFa1",
        "mode": "list",
        "cachedResultName": "All your base",
        "cachedResultUrl": "https://airtable.com/apprtKkVasbQDbFa1"
    },
    "table": {
        "__rl": true,
        "value": "tblInZ7jeNdlU0vxZ",
        "mode": "list",
        "cachedResultName": "Course L2, Workflow 1",
        "cachedResultUrl": "https://airtable.com/apprtKkVasbQDbFa1/tblInZ7jeNdlU0vxZ"
    },
    "options": {}
},
{
"parameters": {
    "id": "e5ae1927-b531-401c-9cb2-ecf1f2836ba6",
    "name": "Airtable",
    "type": "n8n-nodes-base.airtable",
    "typeVersion": 2,
    "position": [
        1000,
        700
    ],
    "credentials": {
        "airtableTokenApi": {
            "id": "MIplo6lY3AEsdf7L",
            "name": "Airtable Personal Access Token account 4"
        }
    },
    {
"parameters": {
        "mode": "combine",
        "mergeByFields": {
            "values": [

```

```

        {
          "field1": "customerID",
          "field2": "customerID"
        }
      ]
    },
    "options": {}
  },
  "id": "1cddc984-7fca-45e0-83b8-0c502cb4c78c",
  "name": "Merge",
  "type": "n8n-nodes-base.merge",
  "typeVersion": 2.1,
  "position": [
    1220,
    600
  ]
},
{
  "parameters": {
    "sortFieldsUi": {
      "sortField": [
        {
          "fieldName": "orderPrice",
          "order": "descending"
        }
      ]
    },
    "options": {}
  },
  "id": "2f55af2e-f69b-4f61-a9e5-c7eefaad93ba",
  "name": "Sort",
  "type": "n8n-nodes-base.sort",
  "typeVersion": 1,
  "position": [
    1440,
    600
  ]
},
{
  "parameters": {
    "conditions": {
      "options": {
        "caseSensitive": true,
        "leftValue": "",
        "typeValidation": "strict"
      },
      "conditions": [
        {
          "id": "d3afe65c-7c80-4caa-9d1c-33c62fbc2197",
          "leftValue": "{{ $json.region }}",
          "rightValue": "Americas",
          "operator": {
            "type": "string",
            "operation": "equals",
            "name": "filter.operator.equals"
          }
        }
      ],
      "combinator": "and"
    },
  }
}

```

```

        "options": {}
    },
    "id": "2ed874a9-5bcf-4cc9-9b52-ea503a562892",
    "name": "If",
    "type": "n8n-nodes-base.if",
    "typeVersion": 2,
    "position": [
        1660,
        500
    ],
    {
        "parameters": {
            "operation": "toJson",
            "mode": "each",
            "options": {
                "fileName": "=report_orderID_{{ ${'If'}).item.json.orderID
}}.json"
            }
        },
        "id": "d93b4429-2200-4a84-8505-16266fedfccd",
        "name": "Convert to File",
        "type": "n8n-nodes-base.convertToFile",
        "typeVersion": 1.1,
        "position": [
            1880,
            500
        ],
        {
            "parameters": {
                "authentication": "webhook",
                "content": "I sent the file using email with the label ID  
and wrote the binary file {file name}. My ID: 123",
                "options": {}
            },
            "id": "26f43f2c-1422-40de-9f40-dd2d80926b1c",
            "name": "Discord",
            "type": "n8n-nodes-base.discord",
            "typeVersion": 2,
            "position": [
                2320,
                500
            ],
            "credentials": {
                "discordWebhookApi": {
                    "id": "WEBrtPdoLrhldYKr",
                    "name": "L2 Course Discord Webhook account"
                }
            },
            {
                "parameters": {
                    "batchSize": 5,
                    "options": {}
                },
                "id": "0fa1fbf6-fe77-4044-a445-c49a1db37dec",
                "name": "Loop Over Items",
                "type": "n8n-nodes-base.splitInBatches",
                "typeVersion": 3,

```

```

"position": [
    1660,
    700
]
},
{
"parameters": {
    "assignments": {
        "assignments": [
            {
                "id": "ce839b80-c50d-48f5-9a24-bb2df6fdd2ff",
                "name": "customerEmail",
                "value": "{ $json.customerEmail }",
                "type": "string"
            },
            {
                "id": "0c613366-3808-45a2-89cc-b34c7b9f3fb7",
                "name": "region",
                "value": "{ $json.region }",
                "type": "string"
            },
            {
                "id": "0f19a88c-deb0-4119-8965-06ed62a840b2",
                "name": "customerSince",
                "value": "{ $json.customerSince }",
                "type": "string"
            },
            {
                "id": "a7e890d6-86af-4839-b5df-d2a4efe923f7",
                "name": "orderPrice",
                "value": "{ $json.orderPrice }",
                "type": "number"
            }
        ]
    },
    "options": {}
},
"id": "09b8584c-4ead-4007-a6cd-edaa4669a757",
"name": "Edit Fields",
"type": "n8n-nodes-base.set",
"typeVersion": 3.3,
"position": [
    1880,
    700
]
},
{
"parameters": {
    "operation": "formatDate",
    "date": "{ $json.customerSince }",
    "options": {
        "includeInputFields": true
    }
},
"id": "c96fae90-e080-48dd-9bff-3e4506aafb86",
"name": "Date & Time",
"type": "n8n-nodes-base.dateTime",
"typeVersion": 2,
"position": [
    2100,

```

```

    700
]
},
{
"parameters": {
    "options": {
        "fileName": "{$runIndex > 0 ? 'file_low_orders':'file_high_orders'}"
    }
},
"id": "43dc8634-2f16-442b-a754-89f47c51c591",
"name": "Convert to File1",
"type": "n8n-nodes-base.convertToFile",
"typeVersion": 1.1,
"position": [
    2320,
    700
]
},
{
"parameters": {
    "authentication": "webhook",
    "content": "I created the spreadsheet {file name}. My ID: 123",
    "options": {}
},
"id": "05dal1c22-d1f6-4ea6-9102-f74f9ae2e9d3",
"name": "Discord1",
"type": "n8n-nodes-base.discord",
"typeVersion": 2,
"position": [
    2540,
    700
],
"credentials": {
    "discordWebhookApi": {
        "id": "WEBrtPdoLrhlDYKr",
        "name": "L2 Course Discord Webhook account"
    }
}
},
],
"connections": {
    "Gmail": {
        "main": [
            [
                {
                    "node": "Discord",
                    "type": "main",
                    "index": 0
                }
            ]
        ],
        "When clicking \"Execute workflow\": {
            "main": [
                [
                    {
                        "node": "HTTP Request",
                        "type": "main",

```

```
        "index": 0
    },
    {
        "node": "Airtable",
        "type": "main",
        "index": 0
    }
]
},
"HTTP Request": {
    "main": [
        [
            [
                {
                    "node": "Merge",
                    "type": "main",
                    "index": 0
                }
            ]
        ]
    ],
    "Airtable": {
        "main": [
            [
                [
                    {
                        "node": "Merge",
                        "type": "main",
                        "index": 1
                    }
                ]
            ]
        ],
        "Merge": {
            "main": [
                [
                    [
                        {
                            "node": "Sort",
                            "type": "main",
                            "index": 0
                        }
                    ]
                ]
            ],
            "Sort": {
                "main": [
                    [
                        [
                            {
                                "node": "Loop Over Items",
                                "type": "main",
                                "index": 0
                            },
                            {
                                "node": "If",
                                "type": "main",
                                "index": 0
                            }
                        ]
                    ]
                ],
                "If": {

```

```
"main": [
  [
    {
      "node": "Convert to File",
      "type": "main",
      "index": 0
    }
  ]
],
"Convert to File": {
  "main": [
    [
      {
        "node": "Gmail",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Loop Over Items": {
  "main": [
    null,
    [
      {
        "node": "Edit Fields",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Edit Fields": {
  "main": [
    [
      {
        "node": "Date & Time",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Date & Time": {
  "main": [
    [
      {
        "node": "Convert to File1",
        "type": "main",
        "index": 0
      }
    ]
  ]
},
"Convert to File1": {
  "main": [
    [
      {
        "node": "Discord1",
        "type": "main",
        "index": 0
      }
    ]
  ]
}
```

```

        "type": "main",
        "index": 0
    }
]
},
"Discord1": {
    "main": [
        [
            {
                "node": "Loop Over Items",
                "type": "main",
                "index": 0
            }
        ]
    ]
},
"pinData": {}
}
```

```

---

## Workflow 3: Monitoring workflow errors

Last but not least, let's help Nathan know if there are any errors running the workflow.

To accomplish this task, create an Error workflow that monitors the main workflow:

1. Create a new workflow.
2. Add an **Error Trigger node** (and execute it as a test).
3. Connect a **Discord node** to the **Error Trigger node** and configure these fields:
  - **Webhook URL:** The Discord URL that you received in the email from n8n when you signed up for this course.
  - **Text:** “The workflow {workflow name} failed, with the error message: {execution error message}. Last node executed: {name of the last executed node}. Check this workflow execution here: {execution URL} My Unique ID:” followed by the unique ID emailed to you when you registered for this course.

Note that you need to replace the text in curly brackets {} with expressions that take the respective information from the Error Trigger node.

4. Execute the Discord node.
5. Set the newly created workflow as the **Error Workflow** for the main workflow you created in the previous lesson.

The workflow should look like this:

### Workflow 3 for monitoring workflow errors



*Workflow 3 for monitoring workflow errors*

```
/// question | Quiz questions * What fields does the Error Trigger node return? * What information about the execution does the Error Trigger node return? * What information about the workflow does the Error Trigger node return? * What's the expression to reference the workflow name? ///
```

---

## Test your knowledge

Congratulations, you finished the n8n Course Level 2!

You've learned a lot about workflow automation and built quite a complex business workflow. Why not showcase your skills?

You can test your knowledge by taking a **quiz**, which consists of questions about the theoretical concepts and workflows covered in this course.

- You need to have at least 80% correct answers to pass the quiz.
- You can take the quiz as many times as you want.
- There's no time limit on answering the quiz questions.

[Take the quiz!](#)

## What's next?

- Create new workflows for your work or personal use and share them with us. Don't have any ideas? Find inspiration on the

[workflows page](#) and on our [blog](#).

- Dive deeper into n8n's features by reading the [docs](#).
- 

## Workflows

A [workflow](#) is a collection of nodes connected together to automate a process.

- [Create](#) a workflow.
- Use [Workflow templates](#) to help you get started.
- Learn about the key [components](#) of an automation in n8n.
- Debug using the [Executions](#) list.
- [Share](#) workflows between users.

If it's your first time building a workflow, you may want to use the [quickstart guides](#) to quickly try out n8n features.

---

## Create a workflow

A [workflow](#) is a collection of nodes connected together to automate a process. You build workflows on the [workflow canvas](#).

### Create a workflow

1. Select the Image: universal create resource icon **button** in the upper-left corner of the side menu. Select workflow.
2. If your n8n instance supports projects, you'll also need to choose whether to create the workflow inside your **personal space** or a specific **project** you have access to. If you're using the community version, you'll always create workflows inside your personal space.
3. Get started by adding a trigger node: select **Add first step...**

Or:

1. Select the Image: universal create resource icon **create** button in the upper-right corner from either the **Overview** page or a specific **project**. Select workflow.
2. If you're doing this from the **Overview** page, you'll create the workflow inside your personal space. If you're doing this from inside a project, you'll create the workflow inside that specific project.
3. Get started by adding a trigger node: select **Add first step...**

If it's your first time building a workflow, you may want to use the [quickstart guides](#) to quickly try out n8n features.

## Run workflows manually

You may need to run your workflow manually when building and testing, or if your workflow doesn't have a trigger node.

To run manually, select **Execute Workflow**.

## Run workflows automatically

All new workflows are unpublished by default. See [Publishing and saving workflows](#).

You need to publish workflows that start with a trigger node or Webhook node so that they can run automatically. When a workflow is inactive, you must run it manually.

To publish your workflow, open your workflow and click **Publish**. The option to unpublish is in the workflow settings menu.

Published workflows run whenever its trigger conditions are met.

---

## Saving and publishing workflows

You can iterate on, test, and save changes to workflows as many times as you need. Then when you're ready to put the workflow into production, publish your workflow.

This approach prevents accidental production changes while enabling safe iteration and review.

Publishing makes your workflow live and locks it to a specific version. Production executions will use this published version, not your latest edits. When you publish, your workflow will enable the following:

- Webhook and form triggers will use their production URLs
- Schedules will run at the times you've defined
- Events from connected apps will trigger this workflow

### Initial state: You open the workflow

[Image: Initial state when workflow is opened]

### Changed state: You make changes to the workflow

[Image: Changes to the workflow]

### Saved state: Saved changes awaiting publish

[Image: Saved changes awaiting publish]

## Checking publishing status

On the **Workflows** page, each workflow displays an indicator showing whether it is **Published** or **Not Published**.

## Publishing a workflow

The **Publish** button in the header indicates that a publish action is pending for one of these reasons:

- When you open a workflow and make changes to it
- When you have saved changes that aren't published

This indicator is independent from the act of saving your changes, which you must still do manually. Each time you save changes to a workflow, n8n creates a new version of the workflow.

You can save your changes as many times as you need to, but the saved version goes live in production only when you publish the workflow.

1. Click the **Publish** button to open the publishing modal
2. The version name defaults to a UUID. Customize the name if you'd like and add a description of the version.
3. Click **Publish** to make your changes live in production. Production executions always point to the currently published version.

If you only update workflow settings, n8n will re-publish the version without requiring you to take any action.

## Managing version history

View and manage version history by clicking the history icon in the header. In the version history view, you can perform these actions:

- Unpublish the workflow to remove it from production
- Restore a previous version. Restoring lets you work on a version without affecting the production execution.
- Publish another version of the workflow

## Unpublishing a Workflow

Unpublish a workflow from either:

- The workflow settings menu
  - The version history page (unpublish action on published versions)
- 

## Workflow components

This section contains:

- Nodes: integrations and operations.
  - Connections: node connectors.
  - Sticky notes: document your workflows.
- 

## Nodes

Nodes are the key building blocks of a workflow. They perform a range of actions, including:

- Starting the workflow.
- Fetching and sending data.
- Processing and manipulating data.

n8n provides a collection of built-in nodes, as well as the ability to create your own nodes. Refer to:

- [Built-in integrations](#) to browse the node library.
- [Community nodes](#) for guidance on finding and installing community-created nodes.
- [Creating nodes](#) to start building your own nodes.

## Add a node to your workflow

### Add a node to an empty workflow

1. Select **Add first step**. n8n opens the nodes panel, where you can search or browse [trigger nodes](#).
2. Select the trigger you want to use.

### Add a node to an existing workflow

Select the **Add node** Image: Add node icon connector. n8n opens the nodes panel, where you can search or browse all nodes.

-8<- “\_snippets/integrations/builtin/node-operations.md”

## Node controls

To view node controls, hover over the node on the canvas:

- **Execute step** Image: Execute step icon: Run the node.
- **Deactivate** Image: Deactivate node icon: Deactivate the node.
- **Delete** Image: Delete node icon: Delete the node.
- **Node context menu** Image: Node context menu icon: Select node actions. Available actions:
  - Open node
  - Execute step
  - Rename node
  - Deactivate node
  - Pin node
  - Copy node
  - Duplicate node
  - Tidy up workflow
  - Convert node to sub-workflow
  - Select all
  - Clear selection
  - Delete node

## Node settings

The node settings under the **Settings** tab allow you to control node behaviors and add node notes.

When active or set, they do the following:

- **Always Output Data**: The node returns an empty item even if the node returns no data during execution. Be careful setting this on IF nodes, as it could cause an infinite loop.
- **Execute Once**: The node executes once, with data from the first

item it receives. It doesn't process any extra items.

- **Retry On Fail:** When an execution fails, the node reruns until it succeeds.
- **On Error:**
  - **Stop Workflow:** Halts the entire workflow when an error occurs, preventing further node execution.
  - **Continue:** Proceeds to the next node despite the error, using the last valid data.
  - **Continue (using error output):** Continues workflow execution, passing error information to the next node for potential handling.

You can document your workflow using node notes:

- **Notes:** Note to save with the node.
  - **Display note in flow:** If active, n8n displays the note in the workflow as a subtitle.
- 

## Connections

A connection establishes a link between nodes to route data through the workflow. A connection between two nodes passes data from one node's output to another node's input.

[Image: Example of creating and deleting a connection]

### Create a connection

To create a connection between two nodes, select the grey dot or **Add node** Image: Add node icon on the right side of a node and slide the arrow to the grey rectangle on the left side of the following node.

### Delete a connection

Hover over the connection, then select **Delete** Image: Delete connector icon.

---

## Sticky Notes

Sticky Notes allow you to annotate and comment on your workflows.

n8n recommends using Sticky Notes heavily, especially on template workflows, to help other users understand your workflow.

[Image: Screenshot of a basic workflow with an example sticky note]

### Create a Sticky Note

Sticky Notes are a core node. To add a new Sticky Note:

1. Open the nodes panel.

2. Search for note.
3. Click the **Sticky Note** node. n8n adds a new Sticky Note to the canvas.

## Edit a Sticky Note

1. Double click the Sticky Note you want to edit.
2. Write your note. [This guide](#) explains how to format your text with Markdown. n8n uses [markdown-it](#), which implements the CommonMark specification.
3. Click away from the note, or press Esc, to stop editing.

## Change the color

To change the Sticky Note color:

1. Hover over the Sticky Note
2. Select **Change color** Image: Change Sticky Note color icon

## Sticky Note positioning

You can:

- Drag a Sticky Note anywhere on the canvas.
- Drag Sticky Notes behind nodes. You can use this to visually group nodes.
- Resize Sticky Notes by hovering over the edge of the note and dragging to resize.
- Change the color: select **Options** Image: Options icon to open the color selector.

## Writing in Markdown

Sticky Notes support Markdown formatting. This section describes some common options.

The text in double asterisks will be **bold**

The text in single asterisks will be **italic**

Use # to indicate headings:

```
This is a top-level heading
This is a sub-heading
This is a smaller sub-heading
```

You can add links:

[Example](<https://example.com/>)

Create lists with asterisks:

```
* Item one
* Item two
```

Or created ordered lists with numbers:

1. Item one
2. Item two

For a more detailed guide, refer to [CommonMark's help](#). n8n uses [markdown-it](#), which implements the CommonMark specification.

## Make images full width

You can force images to be 100% width of the sticky note by appending `#full-width` to the filename:

[\[Image: Source example\]](#)

## Embed a YouTube video

To display a YouTube video in a note, use the `@[youtube](<video-id>)` directive with the video's ID. For this to work, the video's creator must allow embedding.

For example:

`@[youtube](ZCuL2e4zC_4)`

To embed your own video, copy the above syntax, replacing `ZCuL2e4zC_4` with your video ID. The YouTube video ID is the string that follows `v=` in the YouTube URL.

---

## Executions

An execution is a single run of a workflow.

## Execution modes

There are two execution modes:

- Manual: run workflows manually when testing. Select **Execute Workflow** to start a manual execution. You can do manual executions of active workflows, but n8n recommends keeping your workflow set to **Inactive** while developing and testing.
- Production: a production workflow is one that runs automatically. To enable this, set the workflow to **Active**.

## Execution lists

n8n provides two execution lists:

- Workflow-level executions: this execution list shows the executions for a single workflow.
- All executions: this list shows all executions for all your workflows.

n8n supports [adding custom data to executions](#).

---

# Manual, partial, and production executions

There are some important differences in how n8n executes workflows manually (by clicking the **Execute Workflow** button) and automatically (when the workflow is **Active** and triggered by an event or schedule).

## Manual executions

Manual executions allow you to run workflows directly from the canvas to test your workflow logic. These executions are “ad-hoc”: they run only when you manually select the **Execute workflow** button.

Manual executions make building workflows easier by allowing you to iteratively test as you go, following the flow logic and seeing data transformations. You can test conditional branching, data formatting changes, and loop behavior by providing different input items and modifying node options.

## Partial executions

Clicking the **Execute workflow** button at the bottom of the workflow in the **Editor** tab manually runs the entire workflow. You can also perform partial executions to run specific steps in your workflow. Partial executions are manual executions that only run a subset of your workflow nodes.

To perform a partial execution, select a node, open its detail view, and select **Execute step**. This executes the specific node and any preceding nodes required to fill in its input data. You can also temporarily disable specific nodes in the workflow chain to avoid interacting with those services while building.

In particular, partial executions are useful when updating the logic of a specific node since they allow you to re-execute the node with the same input data.

## Troubleshooting partial executions

Some common issues you might come across when running partial executions include the following:

The destination node is not connected to any trigger. Partial executions need a trigger.

This error message appears when you try to perform a partial execution without connecting the workflow to a trigger. Manual executions, including partial executions, attempt to mimic production executions when possible. Part of this includes requiring a trigger node to describe when the workflow logic should execute.

To work around this, connect a trigger node to the workflow with the node you’re trying to execute. Most often, a manual trigger is the simplest option.

Please execute the whole workflow, rather than just the node.  
(Existing execution data is too large.)

This error can appear when performing partial executions on workflows with large numbers of branches. Partial executions involve sending data and workflow logic to the n8n backend in a way that isn't required for full executions. This error occurs when your workflow exceeds the maximum size allowed for these messages.

To work around this, consider using the [limit node](#) to limit node output while running partial executions. Once the workflow is running as intended, you can disable or delete the limit node before enabling production execution.

## Production executions

Production executions occur when a triggering event or schedule automatically runs a workflow.

To configure production executions, you must attach a [trigger node](#) (any trigger other than the [manual trigger](#) works) and switch workflow's toggle to **Active**. Once published, the workflow automatically executes whenever the trigger condition occurs.

The execution flow for production executions doesn't display in the Editor tab of the workflow as with manual executions. Instead, you can see executions in the workflow's **Executions** tab according to your [workflow settings](#). From there, you can explore and troubleshoot problems using the [debug in editor feature](#).

---

## Dirty nodes

A **dirty node** is a node that executed successfully in the past, but whose output n8n now considers stale or unreliable. They're labeled like this to indicate that if the node executes again, the output may be different. It may also be the point where a [partial execution](#) starts from.

### How to recognize dirty node data

In the canvas of the workflow editor, you can identify dirty notes by their different-colored border and a yellow triangle in place of the previous green tick symbol. For example:

[Image: "Image of node displayed with yellow border"]

In the node editor view, the output panel also displays a yellow triangle on the output panel. If you hover over the triangle, a tooltip appears with more information about why n8n considers the data stale:

[Image: "Image of node displayed with yellow border"]

### Why n8n marks nodes dirty

There are several reasons why n8n might flag execution data as stale. For example:

- Inserting or deleting a node: labels the first node that follows the inserted node dirty.
- Modifying node parameters: labels the modified node dirty.
- Adding a connector: labels the destination node of the new connector dirty.
- Deactivating a node: labels the first node that follows the deactivated node dirty.

??? explanation “Other reasons n8n marks nodes dirty” - Unpinning a node: labels the unpinned node dirty. - Modifying pinned data: labels the node that comes after the pinned data dirty. - If any of the above actions occur inside a loop, also labels the first node of the loop dirty.

For sub-nodes, also labels any executed parent nodes (up to and including the root) when:

- Editing an executed sub-node
  - Adding a new sub-node
  - Disconnecting or deleting a sub-node
  - Deactivating a sub-node
  - Activating a sub-node
- When deleting a connected node in a workflow:  
[Image: “Image of node displayed with yellow border”]
  - The next node in the sequence becomes dirty:  
[Image: “Image of node displayed with yellow border”]

When using loops (with the Loop over Items node), when any node within the loop is dirty, the initial node of the loop is also considered dirty:

[Image: “Image of node displayed with yellow border”]

## Resolving dirty nodes

Executing a node again clears its dirty status. You can do this manually by triggering the whole workflow, or by running a partial execution with **Execute step** on the individual node or any node which follows it.

---

## Workflow-level executions list

The **Executions** list in a workflow shows all executions for that workflow.

### View executions for a single workflow

In the workflow, select the **Executions** tab in the top menu. You can preview all executions of that workflow.

## Filter executions

You can filter the executions list.

1. In your workflow, select **Executions**.
2. Select **Filters**.
3. Enter your filters. You can filter by:
  - **Status:** choose from **Failed**, **Running**, **Success**, or **Waiting**.
  - **Execution start:** see executions that started in the given time.
  - **Saved custom data:** this is data you create within the workflow using the Code node. Enter the key and value to filter. Refer to [Custom executions data](#) for information on adding custom data.  
-8<- “\_snippets/workflows/executions/custom-execution-data-availability.md”

## Retry failed workflows

If your workflow execution fails, you can retry the execution. To retry a failed workflow:

1. Open the **Executions** list.
  2. For the workflow execution you want to retry, select **Refresh**  
Image: Refresh icon. -8<- “\_snippets/workflows/executions/retry-options.md”
- 

## All executions

To view **all executions** from an n8n instance, navigate to the **Overview** page and then click into the Executions tab. This will show you all executions from the workflows you have access to.

If your n8n instance supports **projects**, you'll also be able to view the executions tab within projects you have access to. This will show you executions only from the workflows within the specified project.

## Filter executions

You can filter the executions list:

1. Select the **Executions** tab either from within the **Overview** page or a specific **project** to open the list.
2. Select **Filters**.
3. Enter your filters. You can filter by:
  - **Workflows:** choose all workflows, or a specific workflow name.
  - **Status:** choose from **Failed**, **Running**, **Success**, or **Waiting**.
  - **Execution start:** see executions that started in the given time.
  - **Saved custom data:** this is data you create within the workflow using the Code node. Enter the key and value to filter. Refer to [Custom executions data](#) for information on adding custom data.

-8<- “\_snippets/workflows/executions/custom-execution-data-availability.md”

## Retry failed workflows

If your workflow execution fails, you can retry the execution. To retry a failed workflow:

1. Select the **Executions** tab from within either the **Overview** page or a specific **project** to open the list.
2. On the execution you want to retry, select **Retry execution** Image: Options menu icon. -8<- “\_snippets/workflows/executions/retry-options.md”

## Load data from previous executions into your current workflow

You can load data from a previous workflow back into the canvas. Refer to [Debug executions](#) for more information.

---

## Custom executions data

You can set custom data on your workflow using the Code node or the [Execution Data node](#). n8n records this with each execution. You can then use this data when filtering the executions list, or fetch it in your workflows using the Code node.

-8<- “\_snippets/workflows/executions/custom-execution-data-availability.md”

## Set and access custom data using the Code node

This section describes how to set and access data using the Code node. Refer to [Execution Data node](#) for information on using the Execution Data node to set data. You can't retrieve custom data using the Execution Data node.

### Set custom executions data

Set a single piece of extra data:

```
==== "JavaScript" js $execution.customData.set("key", "value");
==== "Python" python _execution.customData.set("key", "value");
```

Set all extra data. This overwrites the whole custom data object for this execution:

```
==== "JavaScript" js $execution.customData.setAll({"key1": "value1", "key2": "value2"})
==== "Python" python _execution.customData.setAll({"key1": "value1", "key2": "value2"})
```

There are limitations:

- They must be strings
- key has a maximum length of 50 characters
- value has a maximum length of 255 characters
- n8n supports a maximum of 10 items of custom data

## Access the custom data object during execution

You can retrieve the custom data object, or a specific value in it, during an execution:

```
==== "JavaScript" ``js // Access the current state of the object during
the execution const customData = $execution.customData.getAll();

// Access a specific value set during this execution
const customData = $execution.customData.get("key");
```

==== "Python" ``python # Access the current state of the object
during the execution customData = _execution.customData.getAll();

# Access a specific value set during this execution
customData = _execution.customData.get("key");
```
```

---

## Debug and re-run past executions

You can load data from a previous execution into your current workflow. This is useful for debugging data from failed production executions: you can see a failed execution, make changes to your workflow to fix it, then re-run it with the previous execution data.

## Load data

To load data from a previous execution:

1. In your workflow, select the **Executions** tab to view the **Executions** list.
  2. Select the execution you want to debug. n8n displays options depending on whether the workflow was successful or failed:
    - For failed executions: select **Debug in editor**.
    - For successful executions: select **Copy to editor**.
  3. n8n copies the execution data into your current workflow, and pins the data in the first node in the workflow.
- 

## Tags

Workflow tags allow you to label your workflows. You can then filter workflows by tag.

Tags are global. This means when you create a tag, it's available to all users on your n8n instance.

## Add a tag to a workflow

To add a tag to your workflow:

1. In your workflow, select **+ Add tag**.
2. Select an existing tag, or enter a new tag name.
3. Once you select a tag and click away from the tag modal, n8n displays the tag next to the workflow name.

You can add more than one tag.

## Filter by tag

When browsing the workflows on your instance, you can filter by tag.

1. On the **Workflows** page, select **Filters**.
2. Select **Tags**.
3. Select the tag or tags you want to filter by. n8n lists the workflows with that tag.

## Manage tags

You can edit existing tags. Instance owners can delete tags.

1. Select **Manage tags**. This is available from **Filters > Tags** on the **Workflows** page, or in the **+ Add tag** modal in your workflow.
  2. Hover over the tag you want to change.
  3. Select **Edit** Image: Add node icon to rename it, or **Delete** Image: Add node icon to delete it.
- 

## Export and import workflows

n8n saves workflows in JSON format. You can export your workflows as JSON files or import JSON files into your n8n library. You can export and import workflows in several ways.

-8<- “\_snippets/workflows/sharing-credentials.md”

## Copy-Paste

You can copy and paste a workflow or parts of it by selecting the nodes you want to copy to the clipboard (Ctrl + c or cmd +c) and pasting it (Ctrl + v or cmd + v) into the Editor UI.

To select all nodes or a group of nodes, click and drag: [Image: Select a group of nodes]

## From the Editor UI menu

From the top navigation bar, select the three dots in the upper right  to see the following options:

## Import/Export menu



### *Import & Export workflows menu*

- **Download:** Downloads your current workflow as a JSON file to your computer.
- **Import from URL:** Imports workflow JSON from a URL, for example, [this workflow JSON file on GitHub](#).
- **Import from File:** Imports a workflow as a JSON file from your computer.

## From the command line

- Export: See the [full list of commands](#) for exporting workflows or credentials.
  - Import: See the [full list of commands](#) for importing workflows or credentials.
- 

## Workflow templates

When creating a new workflow, you can choose whether to start with an empty workflow, or use an existing [template](#).

Templates provide:

- Help getting started: n8n might already have a template that does what you need.
- Examples of what you can build
- Best practices for creating your own workflows

## Access templates

Select Image: View templates icon **Templates** to view the templates library.

If you use n8n's template library, this takes you to browse [Workflows on the n8n website](#). If you use a custom library provided by your organization, you'll be able to search and browse the templates within the app.

## Add your workflow to the n8n library

-8<- “\_snippets/workflows/templates/submit-templates.md”

## Self-hosted n8n: Use your own library

-8<- “\_snippets/workflows/templates/custom-templates-library.md”

---

# Workflow sharing

Workflow sharing allows you to share workflows between users of the same n8n instance.

Users can share workflows they created. Instance owners, and users with the admin role, can view and share all workflows in the instance. Refer to [Account types](#) for more information about owners and admins.

## Share a workflow

1. Open the workflow you want to share.
2. Select **Share**.
3. In **Add users**, find and select the users you want to share with.
4. Select **Save**.

## View shared workflows

You can browse and search workflows on the **Workflows** list. The workflows in the list depend on the project:

- **Overview** lists all workflows you can access. This includes:
  - Your own workflows.
  - Workflows shared with you.
  - Workflows in projects you're a member of.
  - If you log in as the instance owner or admin: all workflows in the instance.
- Other projects: all workflows in the project.

## Workflow roles and permissions

There are two workflow roles: creator and editor. The creator is the user who created the workflow. Editors are other users with access to the workflow.

You can't change the workflow owner, except when deleting the user.

## Permissions

Permissions	Creator	Editor
View workflow (read-only)	:white_check_mark:	:white_check_mark:
View executions	:white_check_mark:	:white_check_mark:
Update (including tags)	:white_check_mark:	:white_check_mark:
Run	:white_check_mark:	:white_check_mark:
Share	:white_check_mark:	:x:
Export	:white_check_mark:	:white_check_mark:
Delete	:white_check_mark:	:x:

## Node editing restrictions with unshared credentials

Sharing in n8n works on the principle of least privilege. This means that if a user shares a workflow with you, but they don't share their credentials, you can't edit the nodes within the workflow that use those credentials. You can view and run the workflow, and edit nodes that don't use unshared credentials.

Refer to [Credential sharing](#) for guidance on sharing credentials.

# Workflow settings

You can customize workflow behavior for individual workflows using workflow settings.

## Access workflow settings

To open the settings:

1. Open your workflow.
2. Select the **three dots icon** Image: three dots icon in the upper-right corner.
3. Select **Settings**. n8n opens the **Workflow settings** modal.

## Available settings

The following settings are available:

### Execution order

Choose the execution order for multi-branch workflows:

**v1 (recommended)** executes each branch in turn, completing one branch before starting another. n8n orders the branches based on their position on the canvas, from topmost to bottommost. If two branches are at the same height, the leftmost branch executes first.

**v0 (legacy)** executes the first node of each branch, then the second node of each branch, and so on.

### Error Workflow (to notify when this one errors)

Select a workflow to trigger if the current workflow fails. See [error workflows](#) for more details.

### This workflow can be called by

Choose which other workflows can call this workflow.

### Timezone

Sets the timezone for this workflow. The timezone setting is important for the Schedule Trigger node.

You can set your n8n instance's timezone to configure the default timezone workflows use:

- [Set a n8n Cloud instance timezone](#)
- [Configure the timezone for self-hosted instances](#)

If you don't configure the workflow or instance timezone, n8n defaults to the EDT (New York) timezone.

### Save failed production executions

Whether n8n should save failed executions for active workflows.

### Save successful production executions

Whether n8n should save successful executions for active workflows.

### Save manual executions

Whether n8n should save executions for workflows started by the user in the editor.

### Save execution progress

Whether n8n should save execution data for each node.

If set to **Save**, the workflow resumes from where it stopped in case of an error. This may increase latency.

### Timeout Workflow

Whether n8n should cancel the current workflow execution after a certain amount of time elapses.

When enabled, the **Timeout After** option appears. Here, you can set the time (in hours, minutes, and seconds) after which the workflow should timeout. For n8n Cloud users, n8n enforces a maximum available timeout for each plan.

### Estimated time saved

An estimate of the number of minutes each of execution of this workflow saves you.

Setting this lets n8n calculate the amount of time saved for [insights](#).

---

## Streaming responses

Streaming responses let you send data back to users as an AI Agent node generates it. This is useful for chatbots, where you want to show the user the answer as it's generated to provide a better user experience.

You can enable streaming using either:

- The [Chat Trigger](#)
- The [Webhook node](#)

In both cases, set the node's **Response Mode** to **Streaming**.

## Configure nodes for streaming

To stream data, you need to add nodes to the workflow that support streaming output. Not all nodes support this feature.

1. Choose a node that supports streaming, such as:
  - [AI agent](#)
  - [Respond to Webhook](#)
2. You can disable streaming in the options of these nodes. By default, they stream data whenever the executed trigger has its Response Mode set to Streaming response.

## Important information

Keep in mind the following details when configuring streaming responses:

- **Trigger:** Your trigger node must support streaming and have streaming configured. Without this, the workflow behaves according to your response mode settings.
  - **Node configuration:** Even with streaming enabled on the trigger, you need at least one node configured to stream data. Otherwise, your workflow will send no data.
- 

## Workflow history

Use workflow history to view and restore previous versions of your workflows.

## Understand workflow history

n8n creates a new version when you:

- Save your workflow.
- Restore an old version. n8n saves the latest version before restoring.
- Pull from a Git repository using [Source control](#). Note that n8n saves versions to the instance database, not to Git.

## View workflow history

To view a workflow's history:

1. Open the workflow.
2. Select **Workflow history** Image: Workflow history icon. n8n opens a menu showing the saved workflow versions, and a canvas with a preview of the selected version.

## Restore or copy previous versions

You can restore a previous workflow version, or make a copy of it:

1. On the version you want to restore or copy, select **Options** Image: Options icon.
  2. Choose what you want to do:
    - **Restore this version**: replace your current workflow with the selected version.
    - **Clone to new workflow**: create a new workflow based on the selected version.
    - **Open version in new tab**: open a second tab displaying the selected version. Use this to compare versions.
    - **Download**: download the version as JSON.
- 

## Find your workflow ID

Your workflow ID is available in:

- The URL of the open workflow.
  - The workflow settings title.
- 

## Sub-workflow conversion

Use sub-workflow conversion to refactor your workflows into reusable parts. Expressions referencing other nodes are automatically updated and added as parameters in the [Execute Workflow Trigger](#) node.

See [sub-workflows](#) for a general introduction to the concept.

## Selecting nodes for a sub-workflow

To convert part of a workflow to a sub-workflow, you must select the nodes in the original workflow that you want to convert.

Do this by selecting a group of valid nodes. The selection must be continuous and must connect to the rest of the workflow from at most one start node and one end node. The selection must fulfill these conditions:

- Must not include trigger nodes.
- Only a single node in the selection can have incoming connections from nodes *outside* of the selection.
  - That node can have multiple incoming connections, but only a single input *branch* (which means it can't be a Merge node for example).
  - That node can't have incoming connections from other nodes in the selection.
- Only a single node in the selection can have outgoing connections to nodes *outside* of the selection.
  - That node can have multiple outgoing connections, but only a single output branch (it can't be an If node for example).
  - That node can't have outgoing connections to other nodes in the selection.
- The selection must include all nodes between the input and output nodes.

## How to convert part of a workflow to a sub-workflow

Select the desired nodes on the canvas. Right-click the canvas background and select **Convert to sub-workflow**.

## Things to keep in mind

Most sub-workflow conversions work without issues, but there are some caveats and limitations to keep in mind:

- **You must set type constraints for input and output manually:** By default, sub-workflow input and output allow all types. You can set expected types in sub-workflow's Execute Sub-workflow Trigger node and Edit Fields (set) node (labeled as **Return** and only included if the sub-workflow has outputs).
- **Limited support for AI nodes:** When dealing with sub-nodes like AI tools, you must select them all and may need to duplicate any nodes shared with other AI Agents before conversion.
- **Uses v1 execution ordering:** New workflows use v1 execution ordering regardless of the parent workflow's settings - you can change this back in the settings.
- **Accessor functions like `first()`, `last()`, and `all()` require extra care:** Expressions using these functions don't always translate cleanly to a sub-workflow context. n8n may transform them to try to preserve their functionality, but you should check that they work as intended in their new context.

- **The `itemMatching` function requires a fixed index:** You can't use expressions for the index value when using the `itemMatching` function. You must pass it a fixed number.
- 

## Credentials

Credentials are private pieces of information issued by apps and services to authenticate you as a user and allow you to connect and share information between the app or service and the n8n node.

Access the credentials UI by opening the left menu and selecting **Credentials**. n8n lists credentials you created on the **My credentials** tab. The **All credentials** tab shows all credentials you can use, included credentials shared with you by other users.

- [Create and edit credentials](#).
  - Learn about [credential sharing](#).
  - Find information on setting up credentials for your services in the [credentials library](#).
- 

## Create and edit credentials

Credentials are securely stored authentication information used to connect n8n workflows to external services such as APIs, or databases.

### Create a credential

1. Select the Image: universal create resource icon **Create** button in the upper-left corner of the side menu. Select credential.
2. If your n8n instance supports [projects](#), you'll also need to choose whether to create the credential inside your personal space or a specific project you have access to. If you're using the community version, you'll create the credential inside your personal space.
3. Select the app or service you wish to connect to.

Or:

1. Using the Image: universal create resource icon **Create** button in the upper-right corner from either the **Overview** page or a specific project. Select Credential.
2. If you're doing this from the **Overview** page, you'll create the credential inside your personal space. If you're doing this from inside a project, you'll create the credential inside that specific project.
3. Select the app or service you wish to connect to.

You can also create new credential in the credential drop down when editing a node on the workflow editor.

Once in the credential modal, enter the details required by your service. Refer to your service's page in the [credentials library](#) for guidance.

When you save a credential, n8n tests it to confirm it works.

## Expressions in credentials

You can use [expressions](#) to set credentials dynamically as your workflow runs:

1. In your workflow, find the data path containing the credential. This varies depending on the exact parameter names in your data.  
Make sure that the data containing the credential is available in the workflow when you get to the node that needs it.
2. When creating your credential, hover over the field where you want to use an expression.
3. Toggle **Expression** on.
4. Enter your expression.

## Example workflow

### Using the example

-8<- “\_snippets/examples-color-key.md”

---

## Credential sharing

You can share a credential directly with other users to use in their own workflows. Or share a credential in a [project](#) for all members of that project to use. Any users using a shared credential won't be able to view or edit the credential details.

Users can share credentials they created and own. Only project admins can share credentials created in and owned by a project. Instance owners and instance admins can view and share all credentials on an instance.

Refer to [Account types](#) for more information about owners and admins.

In [projects](#), a user's role controls how they can interact with the workflows and credentials associated to the projects they're a member of.

## Share a credential

To share a credential:

1. From the left menu, select either **Overview** or a project.
2. Select **Credentials** to see a list of your credentials.
3. Select the credential you want to share.
4. Select **Sharing**.
5. In the **Share with projects or users** dropdown, browse or search for the user or project with which you want to share your credentials.
6. Select a user or project.
7. Select **Save** to apply the changes.

## Remove access to a credential

To unshare a credential:

1. From the left menu, select either **Overview** or a project.
  2. Select **Credentials** to see a list of your credentials.
  3. Select the credential you want to unshare.
  4. Select **Sharing**.
  5. Select **trash icon**: Trash icon on the user or project you want to remove from the list of shared users and projects.
  6. Select **Save** to apply the changes.
- 

## User management

User management in n8n allows you to invite people to work in your n8n instance. It includes:

- Login and password management
- Adding and removing users
- Three account types: **Owner** and **Member** (and **Admin** for Pro & Enterprise plans)

## Setup guides

This section contains most usage information for user management, and the [Cloud setup guide](#). If you self-host n8n, there are extra steps to configure your n8n instance. Refer to the [Self-hosted guide](#).

This section includes guides to configuring [LDAP](#) and [SAML](#) in n8n.

---

## Set up user management on n8n Cloud

To access user management, upgrade to version 0.195.0 or newer.

### Step one: In-app setup

-8<- “\_snippets/user-management/in-app-setup.md”

### Step two: Invite users

-8<- “\_snippets/user-management/invite-users.md”

---

## Manage users

The **Settings > Users** page shows all users, including ones with pending invitations.

## Delete a user

1. Open the **three-dot menu** for the user you want to delete and select **Delete user**.
2. Confirm you want to delete them.
3. If they're an active user, choose whether to copy their workflow data and credentials to a new user, or permanently delete their workflows and credentials.

## Resend an invitation to a pending user

Click the menu icon by the user, then click **Resend invite**.

---

## Account types

There are three account types: owner, admin, and member. The account type affects the user permissions and access.

Permission	Owner	Admin	Member
Manage own email and password	:white_check_mark:	:white_check_mark:	:white_check_mark:
Manage own workflows	:white_check_mark:	:white_check_mark:	:white_check_mark:
View, create, and use tags	:white_check_mark:	:white_check_mark:	:white_check_mark:
Delete tags	:white_check_mark:	:white_check_mark:	:x:
View and share all workflows	:white_check_mark:	:white_check_mark:	:x:
View, edit, and share all credentials	:white_check_mark:	:white_check_mark:	:x:
Set up and use <u>Source control</u>	:white_check_mark:	:white_check_mark:	:x:
Create projects	:white_check_mark:	:white_check_mark:	:x:
View all projects	:white_check_mark:	:white_check_mark:	:x:
Add and remove users	:white_check_mark:	:white_check_mark:	:x:
Access the			

Cloud dashboard	:white_check_mark:	:x:	:x:
-----------------	--------------------	-----	-----

---

## Role-based access control (RBAC)

RBAC is a way of managing access to workflows and [credentials](#) based on user roles and projects. You group workflows into projects, and user access depends on the user's project role. This section provides guidance on using RBAC in n8n.

---

## RBAC role types

Within projects, there are three user roles: Admin, Editor, and Viewer. These roles control what the user can do in a project. A user can have different roles within different projects.

### Project Admin

A Project Admin role has the highest level of permissions. Project admins can:

- Manage project settings: Change name, delete project.
- Manage project members: Invite members and remove members, change members' roles.
- View, create, update, and delete any workflows, credentials, or executions within a project.

### Project Editor

A Project Editor can view, create, update, and delete any workflows, credentials, or executions within a project.

### Project Viewer

A Project Viewer is effectively a `read-only` role with access to all workflows, credentials, and executions within a project.

Viewers aren't able to manually execute any workflows that exist in a project.

Permission	Admin	Editor	Viewer
View workflows in the project	:white_check_mark:	:white_check_mark:	:white_check_mark:
View credentials in the project	:white_check_mark:	:white_check_mark:	:white_check_mark:

View executions	:white_check_mark:	:white_check_mark:	:white_check_mark:
Edit credentials and workflows	:white_check_mark:	:white_check_mark:	:x:
Add workflows and credentials	:white_check_mark:	:white_check_mark:	:x:
Execute workflows	:white_check_mark:	:white_check_mark:	:x:
Manage members	:white_check_mark:	:x:	:x:
Modify the project	:white_check_mark:	:x:	:x:

Variables and tags aren't affected by RBAC: they're global across the n8n instance.

n8n uses projects to group workflows and credentials, and assigns roles to users in each project. This means that a single user can have different roles in different projects, giving them different levels of access.

## Create a project

Instance owners and instance admins can create projects.

To create a project:

1. Select [Image: Plus icon] **Add project**.
2. Fill out the project settings.
3. Select **Save**.

## Add and remove users in a project

Project admins can add and remove users.

To add a user to a project:

1. Select the project.
2. Select **Project settings**.
3. Under **Project members**, browse for users or search by username or email address.
4. Select the user you want to add.
5. Check the role type and change it if needed.
6. Select **Save**.

To remove a user from a project:

1. Select the project.
2. Select **Project settings**.
3. In the **three-dot menu** for the user you want to remove, select

- Remove user.**  
4. Select **Save**.

## Delete a project

To delete a project:

1. Select the project.
2. Select **Project settings**.
3. Select **Delete project**.
4. Choose what to do with the workflows and credentials. You can select:
  - **Transfer its workflows and credentials to another project**: n8n prompts you to choose a project to move the data to.
  - **Delete its workflows and credentials**: n8n prompts you to confirm that you want to delete all the data in the project.

## Move workflows and credentials between projects or users

Workflow and credential owners can move workflows or credentials (changing ownership) to other users or projects they have access to.

1. Select **Workflow menu** Image: Workflow menu icon or **Credential menu** Image: Workflow menu icon > **Move**.
2. Select the project or user you want to move to.
3. Select **Next**.
4. Confirm you understand the impact of the move: workflows may stop working if the credentials they need aren't available in the target project, and n8n removes any current individual sharing.
5. Select **Confirm move to new project**.

## Using external secrets in projects

To use external secrets in a project, you must have an instance owner or instance admin as a member of the project.

---

## Best practices for user management

This page contains advice on best practices relating to user management in n8n.

## All platforms

- n8n recommends that owners create a member-level account for themselves. Owners can see all workflows, but there is no way to

see who created a particular workflow, so there is a risk of overriding other people's work if you build and edit workflows as an owner.

- Users must be careful not to edit the same workflow simultaneously. It's possible to do it, but the users will overwrite each other's changes.
- To move workflows between accounts, export the workflow as JSON, then import it to the new account. Note that this action loses the workflow history.
- Webhook paths must be unique across the entire instance. This means each webhook path must be unique for all workflows and all users. By default, n8n generates a long random value for the webhook path, but users can edit this to their own custom path. If two users set the same path value:
  - The path works for the first workflow that's run or published.
  - Other workflows will error if they try to run with the same path.

## Self-hosted

If you run n8n behind a reverse proxy, set the following environment variables so that n8n generates emails with the correct URL:

- N8N\_HOST
- N8N\_PORT
- N8N\_PROTOCOL
- N8N\_EDITOR\_BASE\_URL

More information on these variables is available in [Environment variables](#).

---

## Two-factor authentication (2FA)

Two-factor authentication (2FA) adds a second authentication method on top of username and password. This increases account security. n8n supports 2FA using an authenticator app.

### Enable 2FA

You need an authenticator app on your phone.

To enable 2FA in n8n:

1. Go to you **Settings > Personal**.
2. Select **Enable 2FA**. n8n opens a modal with a QR code.
3. Scan the QR code in your authenticator app.
4. Enter the code from your app in **Code from authenticator app**.
5. Select **Continue**. n8n displays recovery codes.
6. Save the recovery codes. You need these to regain access to your account if you lose your authenticator.

### Disable 2FA for your instance

Self-hosted users can configure their n8n instance to disable 2FA for all users by setting `N8N_MFA_ENABLED` to false. Note that n8n ignores this if existing users have 2FA enabled. Refer to [Configuration methods](#) for more information on configuring your n8n instance with environment variables.

---

## Lightweight Directory Access Protocol (LDAP)

This page tells you how to enable LDAP in n8n. It assumes you're familiar with LDAP, and have an existing LDAP server set up.

LDAP allows users to sign in to n8n with their organization credentials, instead of an n8n login.

### Enable LDAP

1. Log in to n8n as the instance owner.
2. Select **Settings** Image: Settings icon > **LDAP**.
3. Toggle on **Enable LDAP Login**.
4. Complete the fields with details from your LDAP server.
5. Select **Test connection** to check your connection setup, or **Save connection** to create the connection.

After enabling LDAP, anyone on your LDAP server can sign in to the n8n instance, unless you exclude them using the **User Filter** setting.

You can still create non-LDAP users (email users) on the **Settings > Users** page.

### Merging n8n and LDAP accounts

If n8n finds matching accounts (matching emails) for email users and LDAP users, the user must sign in with their LDAP account. n8n instance owner accounts are excluded from this: n8n never converts owner accounts to LDAP users.

### LDAP user accounts in n8n

On first sign in, n8n creates a user account in n8n for the LDAP user.

You must manage user details on the LDAP server, not in n8n. If you update or delete a user on your LDAP server, the n8n account updates at the next scheduled sync, or when the user next tries to log in, whichever happens first.

### Turn LDAP off

To turn LDAP off:

1. Log in to n8n as the instance owner.
2. Select **Settings** Image: Settings icon > **LDAP**.

### 3. Toggle off **Enable LDAP Login**.

If you turn LDAP off, n8n converts existing LDAP users to email users on their next login. The users must reset their password.

---

## OpenID Connect (OIDC)

-8<- “\_snippets/user-management/sso-oidc-availability.md”

This section covers how to enable and manage OpenID Connect (OIDC) for single sign-on (SSO). You can learn more about how OIDC works by visiting [what is OpenID Connect](#) by the OpenID Foundation.

- [Set up OIDC](#): a general guide to setting up OpenID Connect (OIDC) SSO in n8n.
  - [Troubleshooting](#): a list of things to check if you encounter issues with OIDC.
- 

## Set up OIDC

-8<- “\_snippets/user-management/sso-oidc-availability.md”

### Setting up and enabling OIDC

1. In n8n, go to **Settings > SSO**.
2. Under **Select Authentication Protocol**, choose **OIDC** from the dropdown.
3. Copy the **redirect URL** shown (for example, <https://yourworkspace.app.n8n.cloud/rest/sso/oidc/callback>).
4. Set up OIDC with your identity provider (IdP). You'll need to:
  - Create a new OIDC client/application in your IdP.
  - Configure the redirect URL from the previous step.
  - Note down the **Client ID** and **Client Secret** provided by your IdP.
5. In your IdP, locate the **Discovery Endpoint** (also called the well-known configuration endpoint). It typically has the following format: <https://your-idp-domain/.well-known/openid-configuration>
6. In n8n, complete the OIDC configuration:
  - **Discovery Endpoint**: Enter the discovery endpoint URL from your IdP.
  - **Client ID**: Enter the client ID you received when registering your application with your IdP.
  - **Client Secret**: Enter the client secret you received when registering your application with your IdP.
7. Select **Save settings**.
8. Set OIDC to **Activated**.

## Instance and project access provisioning

From version 1.122.2 upwards, n8n supports provisioning the instance role and project roles from your IdP.

You need to add a an additional scope called n8n to your OIDC authorization server.

On this n8n scope, you need to add these two claims:

Name	Data type	Scope	Token type
n8n_instance_role	string	n8n	ID
n8n_projects	string array	n8n	ID

These two need to always be included in the “ID Token” from your authorization server.

Ensure that these two attributes are configured in the user groups of your identity provider that have access to n8n.

Inside the form on the **Settings > SSO** page in n8n, you will find a dropdown labeled “User role provisioning” in the form.

By default this is set to “Manged in n8n”.

You can choose to set it to:

- Instance role
  - Only the instance role of each provisioned user will be read from the n8n\_instance\_role attribute from the SAML response. Project access will still be managed inside n8n only.
  - If there is no value for n8n\_instance\_role configured on your IdP, the global:member role is used as fallback.
- Instance and project roles
  - Both the instance role and project access of each provisioned user will be read from the n8n\_instance\_role and n8n\_projects attributes from the SAML response.

### Configuring n8n\_instance\_role attribute

The n8n\_instance\_role attribute is a simple string configured for a group or user on your IdP.

Supported instance roles are:

- global:member
- global:admin

### Configuring n8n\_projects attribute

The n8n\_projects attribute is a string array configured for a group or user on your IdP.

Each element in this array needs to follow this format: :

For example:

- bHsykgeFirmIhezz:viewer
- 4K3zrg3DvlMF7TB7:editor
- dCjnYuEpYOUBVaNe:admin

For existing access settings at the time of enabling project access provisioning using your IdP, you can find the project IDs in the downloaded CSV file.

When creating a project from scratch, get the project ID from the URL when viewing the project in your browser:

In the URL <your-domain>/projects/VVRWZaq5DRxaf901/workflows for example, the project ID is VVRWZaq5DRxaf901.

## Provider-specific OIDC setup

### Okta

The steps to setup OIDC in Okta are similar as with Auth0 described below.

For Okta, you can download a visual step-by-step guide as PDF:

[n8n-oidc-with-okta.pdf](#)

### Auth0

1. **Create an application in Auth0:**
  - Log in to your Auth0 Dashboard.
  - Go to **Applications > Applications**.
  - Click **Create Application**.
  - Enter a name (for example, “n8n SSO”) and select **Regular Web Applications**.
  - Click **Create**.
2. **Configure the application:**
  - Go to the **Settings** tab of your new application.
  - **Allowed Callback URLs:** Add your n8n redirect URL from **Settings > SSO > OIDC**.
  - **Allowed Web Origins:** Add your n8n base URL (for example, <https://yourworkspace.app.n8n.cloud>).
  - Click **Save Changes**.
3. **Get your credentials:**
  - **Client ID:** Found in the **Settings** tab.
  - **Client Secret:** Found in the **Settings** tab.
  - **Discovery Endpoint:** <https://{{your-auth0-domain}}.auth0.com/.well-known/openid-configuration>.
4. **In n8n, complete the OIDC configuration:**
  - **Discovery Endpoint:** Enter the discovery endpoint URL from Auth0.
  - **Client ID:** Enter the client ID you found in your Auth0 settings.
  - **Client Secret:** Enter the client secret you found in your Auth0 settings.
5. Select **Save settings**.
6. Set OIDC to **Activated**.

## Discovery endpoints reference

- **Google discovery endpoint example:**

<https://accounts.google.com/.well-known/openid-configuration>

- **Microsoft Azure AD discovery endpoint example:**

<https://login.microsoftonline.com/{tenant-id}/v2.0/.well-known/openid-configuration>

- **Auth0 discovery endpoint example:**

<https://{{your-domain}}.auth0.com/.well-known/openid-configuration>

- **Okta discovery endpoint example:**

<https://{{your-domain}}.okta.com/.well-known/openid-configuration>

- **Amazon Cognito discovery endpoint example:**

<https://cognito-idp.{region}.amazonaws.com/{user-pool-id}/.well-known/openid-configuration>

---

## Troubleshooting OIDC SSO

### Known issues

#### State parameter not supported

When using OIDC providers that enforce the use of the state CSRF token parameter, authentication fails with the error:

```
{"code":0,"message":"authorization response from the server is an error"}
```

n8n's current OIDC implementation doesn't handle the state parameter that some OIDC providers send as a security measure against CSRF attacks.

For now, the only work around is to configure your OIDC provider to disable the state parameter if possible.

n8n is working on adding full support for the OIDC state parameter in a future release.

#### PKCE not supported

OIDC providers that require PKCE (Proof Key for Code Exchange) may fail authentication or reject n8n's authorization requests. n8n's current OIDC implementation doesn't support PKCE.

The only work around is to configure your OIDC provider to not require PKCE for the n8n client if this option is available in your providers settings.

n8n plans on adding PKCE support in a future release

---

## Security Assertion Markup

# Language (SAML)

-8<- “\_snippets/user-management/sso-saml-availability.md”

This section tells you how to enable SAML SSO (single sign-on) in n8n. It assumes you’re familiar with SAML. If you’re not, [SAML Explained in Plain English](#) can help you understand how SAML works, and its benefits.

- [Set up SAML](#): a general guide to setting up SAML in n8n, and links to resources for common IdPs.
  - [Okta Workforce Identity SAML setup](#): step-by-step guidance to configuring Okta.
  - [Troubleshooting](#): a list of things to check if you encounter issues.
  - [Managing users with SAML](#): performing user management tasks with SAML enabled.
- 

## Set up SAML

-8<- “\_snippets/user-management/sso-saml-availability.md”

### Enable SAML

1. In n8n, go to **Settings > SSO**.
2. Make a note of the n8n **Redirect URL** and **Entity ID**.
  - **Optional**: If your IdP allows you to set up SAML from imported metadata, navigate to the **Entity ID** URL and save the XML.
  - **Optional**: If you are running n8n behind a load balancer make sure you have `N8N_EDITOR_BASE_URL` configured.
3. Set up SAML with your identity provider (IdP). You need the **Redirect URL** and **Entity ID**. You may also need an email address and name for the IdP user.
4. After completing setup in your IdP, load the metadata XML into n8n. You can use a metadata URL or raw XML:
  - **Metadata URL**: Copy the metadata URL from your IdP into the **Identity Provider Settings** field in n8n.
  - **Raw XML**: Download the metadata XML from your IdP, toggle **Identiy Provider Settings** to **XML**, and then copy the raw XML into **Identity Provider Settings**.
5. Select **Save settings**.
6. Select **Test settings** to check your SAML setup is working.
7. Set SAML 2.0 to **Activated**.

### Generic IdP setup

The steps to configure the IdP vary depending on your chosen IdP. These are some common setup tasks:

- Create an app for n8n in your IdP.
- Map n8n attributes to IdP attributes:

Value (IdP	Name	Name
---------------	------	------

## side) format

User email	URI Reference	<a href="http://schemas.xmlsoap.org/ws/2005/05/identity/c">http://schemas.xmlsoap.org/ws/2005/05/identity/c</a>
User First Name	URI Reference	<a href="http://schemas.xmlsoap.org/ws/2005/05/identity/c">http://schemas.xmlsoap.org/ws/2005/05/identity/c</a>
User Last Name	URI Reference	<a href="http://schemas.xmlsoap.org/ws/2005/05/identity/c">http://schemas.xmlsoap.org/ws/2005/05/identity/c</a>
User Email	URI Reference	<a href="http://schemas.xmlsoap.org/ws/2005/05/identity/c">http://schemas.xmlsoap.org/ws/2005/05/identity/c</a>

## Instance and project access provisioning

From version 1.122.2 upwards, n8n supports provisioning the instance role and project roles from your IdP.

You need to configure these attributes on the groups or individual users in your IdP:

Value (IdP side)	Data type	Name
n8n_instance_role	string	n8n_instance_role
n8n_projects	array	n8n_projects

On the form on the **Settings > SSO** page, located the dropdown labeled **User role provisioning**. n8n sets the default value of this dropdown to **Managed in n8n**.

You can choose to set it to one of these values:

- **Instance role**
  - Only the instance role of each provisioned user is read from the n8n\_instance\_role attribute from the SAML response. Project access is still managed inside n8n only.
  - If there is no value for n8n\_instance\_role configured on your IdP, the global:member role is used as fallback.
- **Instance and project roles**
  - Both the instance role and project access of each provisioned user are read from the n8n\_instance\_role and n8n\_projects attributes from the SAML response.

### Configuring n8n\_instance\_role attribute

The n8n\_instance\_role attribute is a string configured for a group or user on your IdP.

Supported instance roles are:

- global:member
- global:admin

### Configuring n8n\_projects attribute

The n8n\_projects attribute is a string array configured for a group or user on your IdP.

Each element in this array needs to follow this format: <project-id>: <role>

For example:

- bHsykgeFirmIhezz:viewer
- 4K3zrg3DvlMFFTb7:editor
- dCjnYuEpY0UBVaNe:admin

For existing access settings at the time of enabling project access provisioning through your IdP, you can find the project IDs in the downloaded CSV file.

When creating a project from scratch, get the project ID from the URL when viewing the project in your browser:

In the URL <your-domain>/projects/VVRWZaq5DRxaf901/workflows for example, the project ID is VVRWZaq5DRxaf901.

## Setup resources for common IdPs

Documentation links for common IdPs.

IdP	Documentation
Auth0	<a href="#">Configure Auth0 as SAML Identity Provider: Manually configure SSO integrations</a>
Authentik	<a href="#">Applications and the SAML Provider</a>
Azure AD	<a href="#">SAML authentication with Azure Active Directory</a>
JumpCloud	<a href="#">How to setup SAML (SSO) applications with JumpCloud (using Zoom as an example)</a>
Keycloak	Choose a <a href="#">Getting Started</a> guide depending on your hosting.
Okta	n8n provides a <a href="#">Workforce Identity setup guide</a> as well as a <a href="#">step-by-step PDF guide</a>
PingIdentity	<a href="#">PingOne SSO</a>

## Okta Workforce Identity SAML setup

Set up SAML SSO in n8n with Okta.

### Prerequisites

You need an Okta Workforce Identity account, and the redirect URL and entity ID from n8n's SAML settings.

Okta Workforce may enforce two factor authentication for users, depending on your Okta configuration.

Read the [Set up SAML](#) guide first.

## Setup

In addition to the following instructions, [this PDF](#) provides visual step-by-step guide on how to setup SAML in n8n with Okta.

1. In your Okta admin panel, select **Applications > Applications**.
2. Select **Create App Integration**. Okta opens the app creation modal.
3. Select **SAML 2.0**, then select **Next**.
4. On the **General Settings** tab, enter n8n as the **App name**.
5. Select **Next**.
6. On the **Configure SAML** tab, complete the following **General** fields:
  - **Single sign-on URL:** the **Redirect URL** from n8n.
  - **Audience URI (SP Entity ID):** the **Entity ID** from n8n.
  - **Default RelayState:** leave this empty.
  - **Name ID format:** `EmailAddress`.
  - **Application username:** Okta username.
  - **Update application username on:** Create and update.
7. Create **Attribute Statements**:

Name
<code>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/firstname</code>
<code>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/lastname</code>
<code>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/upn</code>
<code>http://schemas.xmlsoap.org/ws/2005/05/identity/claims/emailaddress</code>

8. Select **Next**. Okta may prompt you to complete a marketing form, or may take you directly to your new n8n Okta app.
9. Assign the n8n app to people:
  1. On the n8n app dashboard in Okta, select **Assignments**.
  2. Select **Assign > Assign to People**. Okta displays a modal with a list of available people.
  3. Select **Assign** next to the person you want to add. Okta displays a prompt to confirm the username.
  4. Leave the username as email address. Select **Save and Go Back**.
  5. Select **Done**.
10. Get the metadata XML: on the **Sign On** tab, copy the Metadata URL. Navigate to it, and copy the XML. Paste this into **Identity Provider Settings** in n8n.
11. Select **Save settings**.

12. Select **Test settings**. n8n opens a new tab. If you're not currently logged in, Okta prompts you to sign in. n8n then displays a success message confirming the attributes returned by Okta.

## Instance and project access provisioning

See the [Set up SAML page](#) for additional information of this feature.

### Adding the required attributes

1. In your Okta admin panel, select **Applications > Applications**.
2. Go to the configuration of your n8n application
3. On the **General** tab, click **Edit** next to **SAML Settings**
4. In the page that opens, continue to step 2: **Configure SAML**
5. Add the following two **Attribute Statements**:

Name	Name format	Value
n8n_instance_role	Basic	appuser.n8n_instance_role
n8n_projects	Basic	appuser.n8n_projects

6. Click **Next**

7. Click **Finish**

### Updating the app profile

1. In your Okta admin panel, select **Directory > Profile Editor**.
2. Go to the profile of your n8n application
3. Click **Add Attribute**
4. Add the **n8n\_instance\_role** attribute
  - **Data type**: string
  - **Display name**: n8n\_instance\_role
  - **Variable name**: n8n\_instance\_role
  - **Attribute type**: Group
5. Add the **n8n\_projects** attribute
  - **Data type**: string array
  - **Display name**: n8n\_projects
  - **Variable name**: n8n\_projects
  - **Attribute type**: Group
  - **Group priority**: Combine values across groups

Now when you go to **Directory > Groups** and edit the assigned n8n application, you can configure the **n8n\_instance\_role** and **n8n\_projects** to be sent to n8n upon logging in via SAML.

---

## Manage users with SAML

-8<- “\_snippets/user-management/sso-saml-availability.md”

There are some user management tasks that are affected by SAML.

## Exempt users from SAML

You can allow users to log in without using SAML. To do this:

1. Go to **Settings > Users**.
2. Select the menu icon by the user you want to exempt from SAML.
3. Select **Allow Manual Login**.

## Deleting users

If you remove a user from your IdP, they remain logged in to n8n. You need to manually remove them from n8n as well. Refer to [Manage users](#) for guidance on deleting users.

---

## Troubleshooting SAML SSO

If you get an error when testing your SAML setup, check the following:

- Does the app you created in your IdP support SAML?
- Did you enter the n8n redirect URL and entity ID in the correct fields in your IdP?
- Is the metadata XML correct? Check that the metadata you copied into n8n is formatted correctly.

For more support, use the [forum](#), or contact your support representative if you have a paid support plan.

---

## Keyboard shortcuts and controls

n8n provides keyboard shortcuts for some actions.

## Workflow controls

- **Ctrl + Alt + n**: create new workflow
- **Ctrl + o**: open workflow
- **Ctrl + s**: save the current workflow
- **Ctrl + z**: undo
- **Ctrl + shift + z**: redo
- **Ctrl + Enter**: execute workflow

## Canvas

### Move the canvas

- **Ctrl + Left Mouse Button + drag**: move node view
- **Ctrl + Middle mouse button + drag**: move node view
- **Space + drag**: move node view
- **Middle mouse button + drag**: move node view

- Two fingers on a touch screen: move node view

## Canvas zoom

- **+** or **=**: zoom in
- **-** or **\*\*\_\*\***: zoom out
- **0**: reset zoom level
- **1**: zoom to fit workflow
- **Ctrl + Mouse wheel**: zoom in/out

## Nodes on the canvas

- **Double click** on a node: open the node details
- **Ctrl/Cmd + Double click** on a sub-workflow node: open the sub-workflow in a new tab
- **Ctrl + a**: select all nodes
- **Ctrl + v**: paste nodes
- **Shift + s**: add sticky note

## With one or more nodes selected in canvas

- **ArrowDown**: select sibling node below the current one
- **ArrowLeft**: select node left of the current one
- **ArrowRight**: select node right of the current one
- **ArrowUp**: select sibling node above the current one
- **Ctrl + c**: copy
- **Ctrl + x**: cut
- **D**: deactivate
- **Delete**: delete
- **Enter**: open
- **F2**: rename
- **P**: pin data in node. Refer to [Data pinning](#) for more information.
- **Shift + ArrowLeft**: select all nodes left of the current one
- **Shift + ArrowRight**: select all nodes right of the current one
- **Ctrl/Cmd + Shift + o** on a sub-workflow node: open the sub-workflow in a new tab

## Node panel

- **Tab**: open the Node Panel
- **Enter**: insert selected node into workflow
- **Escape**: close Node panel

## Node panel categories

- **Enter**: insert node into workflow, collapse/expand category, open subcategory
- **ArrowRight**: expand category, open subcategory
- **ArrowLeft**: collapse category, close subcategory view

## Within nodes

- **=**: in an empty parameter input, this switches to [expressions](#) mode.

## Command bar

The Command Bar provides quick access to actions and navigation throughout n8n. Access it using **Ctrl/Cmd + K**, or click the magnifying glass icon on the canvas. Commands adapt based on your current view and permissions.

- **Workflow actions:** Add nodes, save, test, tidy up, publish/unpublish, duplicate, import/export, archive, delete
  - **Resource navigation:** Create and open workflows, credentials, data tables, projects; access recent resources
  - **Execution actions:** Debug, copy, retry, stop, or delete executions
  - **General navigation:** Access Templates, Variables, Insights, Settings, Help resources, and Documentation
- 

## Flow logic

n8n allows you to represent complex logic in your workflows.

This section covers:

### Related sections

You need some understanding of Data in n8n, including Data structure and Data flow within nodes.

When building your logic, you'll use n8n's Core nodes, including:

- Splitting: IF and Switch.
  - Merging: Merge, Compare Datasets, and Code.
  - Looping: IF and Loop Over Items.
  - Waiting: Wait.
  - Creating sub-workflows: Execute Workflow and Execute Workflow Trigger.
  - Error handling: Stop And Error and Error Trigger.
- 

## Splitting workflows with conditional nodes

Splitting uses the IF or Switch nodes. It turns a single-branch workflow into a multi-branch workflow. This is a key piece of representing complex logic in n8n.

Compare these workflows:

[Image: “Diagram representing two workflows. One has three steps and follows a linear process, with a user submitting a bug, and the workflow emailing a support team. The second workflow starts the same way, but then splits depending on whether the user marked the issue as urgent. It then splits again depending on the user’s support plan”]

This is the power of splitting and conditional nodes in n8n.

Refer to the [IF](#) or [Switch](#) documentation for usage details.

---

## Merging data

Merging brings multiple data streams together. You can achieve this using different nodes depending on your workflow requirements.

- Merge data from different data streams or nodes: Use the [Merge](#) node to combine data from various sources into one.
- Merge data from multiple node executions: Use the [Code](#) node for complex scenarios where you need to merge data from multiple executions of a node or multiple nodes.
- Compare and merge data: Use the [Compare Datasets](#) node to compare, merge, and output data streams based on the comparison.

Explore each method in more detail in the sections below.

### Merge data from different data streams

If your workflow [splits](#), you combine the separate streams back into one stream.

Here's an [example workflow](#) showing different types of merging: appending data sets, keeping only new items, and keeping only existing items. The [Merge node](#) documentation contains details on each of the merge operations.

### Merge data from different nodes

You can use the Merge node to combine data from two previous nodes, even if the workflow hasn't split into separate data streams. This can be useful if you want to generate a single dataset from the data generated by multiple nodes.

[Image: Merging data from two previous nodes. The diagram shows three nodes lined up sequentially. The first node is labeled Fetch data, the second is labeled Modify data, and the third is labeled Merge: append both data sets. Arrows connect nodes 1 to 2, 2 to 3, and 1 to 3.]  
Merging data from two previous nodes

### Merge data from multiple node executions

Use the Code node to merge data from multiple node executions. This is useful in some [Looping](#) scenarios.

Refer to this [example workflow](#) using Loop Over Items and Wait to artificially create multiple executions.

### Compare, merge, and split again

The [Compare Datasets](#) node compares data streams before merging them. It outputs up to four different data streams.

Refer to this [example workflow](#) for an example.

---

## Looping in n8n

Looping is useful when you want to process multiple items or perform an action repeatedly, such as sending a message to every contact in your address book. n8n handles this repetitive processing automatically, meaning you don't need to specifically build loops into your workflows. There are [some nodes](#) where this isn't true.

## Using loops in n8n

n8n nodes take any number of items as input, process these items, and output the results. You can think of each item as a single data point, or a single row in the output table of a node.

[Image: The Customer Datastore node output]

Nodes usually run once for each item. For example, if you wanted to send the name and notes of the customers in the Customer Datastore node as a message on Slack, you would:

1. Connect the Slack node to the Customer Datastore node.
2. Configure the parameters.
3. Execute the node.

You would receive five messages: one for each item.

This is how you can process multiple items without having to explicitly connect nodes in a loop.

## Executing nodes once

For situations where you don't want a node to process all received items, for example sending a Slack message only to the first customer, you can do so by toggling the **Execute Once** parameter in the **Settings** tab of that node. This setting is helpful when the incoming data contains multiple items and you want to only process the first one.

## Creating loops

n8n typically handles the iteration for all incoming items. However, there are certain scenarios where you will have to create a loop to iterate through all items. Refer to [Node exceptions](#) for a list of nodes that don't automatically iterate over all incoming items.

### Loop until a condition is met

To create a loop in an n8n workflow, connect the output of one node to the input of a previous node. Add an IF node to check when to stop the loop.

Here is an example workflow that implements a loop with an IF node:

[Image: Editor UI view of sample workflow]

## Loop until all items are processed

Use the Loop Over Items node when you want to loop until all items are processed. To process each item individually, set **Batch Size** to 1.

You can batch the data in groups and process these batches. This approach is useful for avoiding API rate limits when processing large incoming data or when you want to process a specific group of returned items.

The Loop Over Items node stops executing after all the incoming items get divided into batches and passed on to the next node in the workflow so it's not necessary to add an IF node to stop the loop.

## Node exceptions

Nodes and operations where you need to design a loop into your workflow:

- CreateDB executes once for insert and update.
- Code node in **Run Once for All Items** mode: processes all the items based on the entered code snippet.
- Execute Workflow node in **Run Once for All Items** mode.
- HTTP Request: you must handle pagination yourself. If your API call returns paginated results you must create a loop to fetch one page at a time.
- Microsoft SQL executes once for insert, update, and delete.
- MongoDB executes once for insert and update.
- QuestDB executes once for insert.
- Redis:
  - Info: this operation executes only once, regardless of the number of items in the incoming data.
- RSS Read executes once for the requested URL.
- TimescaleDB executes once for insert and update.

---

## Waiting

Waiting allows you to pause a workflow mid-execution, then resume where the workflow left off, with the same data. This is useful if you need to rate limit your calls to a service, or wait for an external event to complete. You can wait for a specified duration, or until a webhook fires.

Making a workflow wait uses the Wait node. Refer to the node documentation for usage details.

n8n provides a workflow template with a basic example of Rate limiting and waiting for external events.

---

## Sub-workflows

You can call one workflow from another workflow. This allows you to build modular, microservice-like workflows. It can also help if your workflow grows large enough to encounter [memory issues](#). Creating sub-workflows uses the [Execute Workflow](#) and [Execute Sub-workflow Trigger](#) nodes.

Sub-workflow executions don't count towards your plan's monthly execution or active workflow limits.

### Set up and use a sub-workflow

This section walks through setting up both the parent workflow and sub-workflow.

```
-8<- "_snippets/flow-logic/subworkflow-usage.md"
```

### How data passes between workflows

```
-8<- "_snippets/flow-logic/subworkflow-data-flow.md"
```

### Sub-workflow conversion

See [sub-workflow conversion](#) for how to divide your existing workflows into sub-workflows.

---

## Error handling

When designing your flow logic, it's a good practice to consider potential errors, and set up methods to handle them gracefully. With an error workflow, you can control how n8n responds to a workflow execution failure.

### Create and set an error workflow

For each workflow, you can set an error workflow in **Workflow Settings**. It runs if an execution fails. This means you can, for example, send email or Slack alerts when a workflow execution errors. The error workflow must start with the [Error Trigger](#).

You can use the same error workflow for multiple workflows.

```
-8<- "_snippets/flow-logic/create-set-error-workflow.md"
```

### Error data

```
-8<- "_snippets/integrations/builtin/core-nodes/error-trigger/error-data.md"
```

## Cause a workflow execution failure using Stop And Error

When you create and set an error workflow, n8n runs it when an execution fails. Usually, this is due to things like errors in node settings, or the workflow running out of memory.

You can add the [Stop And Error](#) node to your workflow to force executions to fail under your chosen circumstances, and trigger the error workflow.

---

## Execution order in multi-branch workflows

n8n's node execution order depends on the version of n8n you're using:

- For workflows created before version 1.0: n8n executes the first node of each branch, then the second node of each branch, and so on.
- For workflows created in version 1.0 and above: executes each branch in turn, completing one branch before starting another. n8n orders the branches based on their position on the [canvas](#), from topmost to bottommost. If two branches are at the same height, the leftmost branch executes first.

You can change the execution order in your [workflow settings](#).

---

## Data

Data is the information that n8n nodes receive and process. For basic usage of n8n you don't need to understand data structures and manipulation. However, it becomes important if you want to:

- Create your own node
- Write custom [expressions](#)
- Use the Function or Function Item node

This section covers:

- [Data structure](#)
- [Data flow within nodes](#)
- [Transforming data](#)
- [Process data using code](#)
- [Pinning](#) and [editing](#) data during workflow development.
- [Data mapping](#) and [Item linking](#): how data items link to each other.

## Related resources

### Data transformation nodes

n8n provides a collection of nodes to transform data:

- Aggregate: take separate items, or portions of them, and group them together into individual items.
  - Limit: remove items beyond a defined maximum number.
  - Remove Duplicates: identify and delete items that are identical across all fields or a subset of fields.
  - Sort: organize lists of in a desired ordering, or generate a random selection.
  - Split Out: separate a single data item containing a list into multiple items.
  - Summarize: aggregate items together, in a manner similar to Excel pivot tables.
- 

## Data structure

In n8n, all data passed between nodes is an array of objects. It has the following structure:

```
[
 {
 // For most data:
 // Wrap each item in another object, with the key 'json'
 "json": {
 // Example data
 "apple": "beets",
 "carrot": {
 "dill": 1
 }
 },
 // For binary data:
 // Wrap each item in another object, with the key 'binary'
 "binary": {
 // Example data
 "apple-picture": {
 "data": "....", // Base64 encoded binary data
 (required) "mimeType": "image/png", // Best practice to set if
 possible (optional) "fileExtension": "png", // Best practice to set if
 possible (optional) "fileName": "example.png", // Best practice to set
 if possible (optional)
 }
 },
 },
]
```

## Data item processing

-8<- “\_snippets/flow-logic/data-flow-nodes.md”

---

## Data flow within nodes

-8<- “\_snippets/flow-logic/data-flow-nodes.md”

---

# Transforming data

n8n uses a predefined [data structure](#) that allows all nodes to process incoming data correctly.

Your incoming data may have a different data structure, in which case you will need to transform it to allow each item to be processed individually.

For example, the image below shows the output of an [HTTP Request](#) node that returns data incompatible with n8n's data structure. The node returns the data and displays that only one item was returned.

[Image: HTTP Request node output]

To transform this kind of structure into the n8n data structure you can use the data transformation nodes:

- [Aggregate](#): take separate items, or portions of them, and group them together into individual items.
  - [Limit](#): remove items beyond a defined maximum number.
  - [Remove Duplicates](#): identify and delete items that are identical across all fields or a subset of fields.
  - [Sort](#): organize lists of in a desired ordering, or generate a random selection.
  - [Split Out](#): separate a single data item containing a list into multiple items.
  - [Summarize](#): aggregate items together, in a manner similar to Excel pivot tables.
- 

# Processing data with code

## Function

A function is a block of code designed to perform a certain task. In n8n, you can write custom JavaScript or Python code snippets to add, remove, and update the data you receive from a node.

The [Code](#) node gives you access to the incoming data and you can manipulate it. With this node you can create any function you want using JavaScript code.

---

# Data mapping

Data mapping means referencing data from previous nodes.

This section contains guidance on:

- Mapping data in most scenarios: [Data mapping in the UI](#) and [Data mapping in expression](#)
  - How to handle [item linking](#) when using the Code node or building your own nodes.
-

# Mapping in the UI

Data mapping means referencing data from previous nodes. It doesn't include changing (transforming) data, just referencing it.

You can map data in the following ways:

- Using the expressions editor.
- By dragging and dropping data from the **INPUT** into parameters. This generates the expression for you.

For information on errors with mapping and linking items, refer to [Item linking errors](#).

## How to drag and drop data

1. Run your workflow to load data.
2. Open the node where you need to map data.
3. You can map in table, JSON, and schema view:
  - In table view: click and hold a table heading to map top level data, or a field in the table to map nested data.
  - In JSON view: click and hold a key.
  - In schema view: click and hold a key.
4. Drag the item into the field where you want to use the data.

## Understand what you're mapping with drag and drop

Data mapping maps the key path, and loads the key's value into the field. For example, given the following data:

```
[
 {
 "fruit": "apples",
 "color": "green"
 }
]
```

You can map **fruit** by dragging and dropping **fruit** from the **INPUT** into the field where you want to use its value. This creates an expression, `{{ $json.fruit }}`. When the node iterates over input items, the value of the field becomes the value of **fruit** for each item.

## Understand nested data

Given the following data:

```
[
 {
 "name": "First item",
 "nested": {
 "example-number-field": 1,
 "example-string-field": "apples"
 }
 },
 {
 "name": "Second item",
 "nested": {
 "example-number-field": 2,
 "example-string-field": "bananas"
 }
 }]
```

```
 "example-string-field": "oranges"
 }
}
]
```

n8n displays it in table form like this:

[Image: “Screenshot of a table in the INPUT panel. It includes a top level field named “nested.” This field contains nested data, which is indicated in bold.”]

---

## Mapping in the expressions editor

These examples show how to access linked items in the expressions editor. Refer to [expressions](#) for more information on expressions, including built in variables and methods.

For information on errors with mapping and linking items, refer to [Item linking errors](#).

### Access the linked item in a previous node’s output

When you use this, n8n works back up the item linking chain, to find the parent item in the given node.

```
// Returns the linked item
{{$(<node-name>).item}}
```

As a longer example, consider a scenario where a node earlier in the workflow has the following output data:

```
[
{
 {
 "id": "23423532",
 "name": "Jay Gatsby",
 },
 {
 "id": "23423533",
 "name": "José Arcadio Buendía",
 },
 {
 "id": "23423534",
 "name": "Max Sendak",
 },
 {
 "id": "23423535",
 "name": "Zaphod Beeblebrox",
 },
 {
 "id": "23423536",
 "name": "Edmund Pevensie",
 }
]
```

To extract the name, use the following expression:

```
{{$(<node-name>).item.json.name}}
```

## Access the linked item in the current node's input

In this case, the item linking is within the node: find the input item that the node links to an output item.

```
// Returns the linked item
{$input.item}}
```

As a longer example, consider a scenario where the current node has the following input data:

```
[
 {
 "id": "23423532",
 "name": "Jay Gatsby",
 },
 {
 "id": "23423533",
 "name": "José Arcadio Buendía",
 },
 {
 "id": "23423534",
 "name": "Max Sendak",
 },
 {
 "id": "23423535",
 "name": "Zaphod Beeblebrox",
 },
 {
 "id": "23423536",
 "name": "Edmund Pevensie",
 }]
```

To extract the name, you'd normally use drag-and-drop [Data mapping](#), but you could also write the following expression:

```
{$input.item.json.name}
```

---

## Data item linking

An item is a single piece of data. Nodes receive one or more items, operate on them, and output new items. Each item links back to previous items.

You need to understand this behavior if you're:

- Building a programmatic-style node that implements complex behaviors with its input and output data.
- Using the Code node or expressions editor to access data from earlier items in the workflow.
- Using the Code node for complex behaviors with input and output data.

This section provides:

- A conceptual overview of [Item linking concepts](#).
  - Information on [Item linking for node creators](#).
  - Support for end users who need to [Work with the data path](#) to retrieve item data from previous nodes, and link items when using the Code node.
  - Guidance on troubleshooting [Errors](#).
- 

## Item linking concepts

Each output item created by a node includes metadata that links them to the input item (or items) that the node used to generate them. This creates a chain of items that you can work back along to access previous items. This can be complicated to understand, especially if the node splits or merges data. You need to understand item linking when building your own programmatic nodes, or in some scenarios using the Code node.

This document provides a conceptual overview of this feature. For usage details, refer to:

- [Item linking for node creators](#), for details on how to handle item linking when building a node.
- [Item linking in the Code node](#), to learn how to handle item linking in the Code node.
- [Item linking errors](#), to understand the errors you may encounter in the editor UI.

## n8n's automatic item linking

If a node doesn't control how to link input items to output items, n8n tries to guess how to link the items automatically:

- Single input, single output: the output links to the input.
- Single input, multiple outputs: all outputs link to that input.
- Multiple inputs and outputs:
  - If you keep the input items, but change the order (or remove some but keep others), n8n can automatically add the correct linked item information.
  - If the number of inputs and outputs is equal, n8n links the items in order. This means that output-1 links to input-1, output-2 to input-2, and so on.
  - If the number isn't equal, or you create completely new items, n8n can't automatically link items.

If n8n can't link items automatically, and the node doesn't handle the item linking, n8n displays an error. Refer to [Item linking errors](#) for more information.

## Item linking example

[Image: A diagram showing the threads linking multiple items back through a workflow]

In this example, it's possible for n8n to link an item in one node back several steps, despite the item order changing. This means the node that sorts movies alphabetically can access information about the

linked item in the node that gets famous movie actors.

The methods for accessing linked items are different depending on whether you're using the UI, expressions, or the code node. Explore the following resources:

- [Mapping in the UI](#)
  - [Mapping in the expressions editor](#)
  - [Item linking in the Code node](#)
  - [Item linking errors](#)
- 

## Item linking in the Code node

-8<- “\_snippets/data/data-mapping/item-linking-code-node.md”

---

## Item linking errors

In n8n you can reference data from any previous node. This doesn't have to be the node just before: it can be any previous node in the chain. When referencing nodes further back, you use the expression syntax `$(node_name).item`.

[Image: A diagram showing the threads linking multiple items back through a workflow]

Diagram of threads for different items. Due to the item linking, you can get the actor for each movie using `$('Get famous movie actors').item`.

Since the previous node can have multiple items in it, n8n needs to know which one to use. When using `.item`, n8n figures this out for you behind the scenes. Refer to [Item linking concepts](#) for detailed information on how this works.

`.item` fails if information is missing. To figure out which item to use, n8n maintains a thread back through the workflow's nodes for each item. For a given item, this thread tells n8n which items in previous nodes generated it. To find the matching item in a given previous node, n8n follows this thread back until it reaches the node in question.

When using `.item`, n8n displays an error when:

- The thread is broken
- The thread points to more than one item in the previous node (as it's unclear which one to use)

To solve these errors, you can either avoid using `.item`, or fix the root cause.

You can avoid `.item` by using `.first()`, `.last()` or `.all()[index]` instead. They require you to know the position of the item that you're targeting within the target node's output items. Refer to [Built in methods and variables | Output of other nodes](#) for more detail on these methods.

The fix for the root cause depends on the exact error.

## **Fix for ‘Info for expressions missing from previous node’**

If you see this error message:

| ERROR: Info for expression missing from previous node

There’s a node in the chain that doesn’t return pairing information. The solution here depends on the type of the previous node:

- Code nodes: make sure you return which input items the node used to produce each output item. Refer to [Item linking in the code node](#) for more information.
- Custom or community nodes: the node creator needs to update the node to return which input items it uses to produce each output item. Refer to [Item linking for node creators](#) for more information.

## **Fix for ‘Multiple matching items for expression’**

This is the error message:

| ERROR: Multiple matching items for expression

Sometimes n8n uses multiple items to create a single item. Examples include the Summarize, Aggregate, and Merge nodes. These nodes can combine information from multiple items.

When you use `.item` and there are multiple possible matches, n8n doesn’t know which one to use. To solve this you can either:

- Use `.first()`, `.last()` or `.all()[index]` instead. Refer to [Built in methods and variables | Output of other nodes](#) for more detail on these methods.
- Reference a different node that contains the same information, but doesn’t have multiple matching items.

---

## **Item linking for node creators**

-8<- “\_snippets/data/data-mapping/item-linking-node-creators.md”

---

## **Data pinning**

You can ‘pin’ data during workflow development. Data pinning means saving the output data of a node, and using the saved data instead of fetching fresh data in future workflow executions.

You can use this when working with data from external sources to avoid having to repeat requests to the external system. This can save time and resources:

- If your workflow relies on an external system to trigger it, such as a webhook call, being able to pin data means you don’t need to use the external system every time you test the workflow.
- If the external resource has data or usage limits, pinning data during tests avoids consuming your resource limits.

- You can fetch and pin the data you want to test, then have confidence that the data is consistent in all your workflow tests.

You can only pin data for nodes that have a single main output (“error” outputs don’t count for this purpose).

## Pin data

-8<- “\_snippets/data/how-to-pin-data.md”

## Unpin data

When data pinning is active, a banner appears at the top of the node’s output panel indicating that n8n has pinned the data. To unpin data and fetch fresh data on the next execution, select the **Unpin** link in the banner.

---

## Data editing

n8n allows you to edit pinned data. This means you can check different scenarios without setting up each scenario and sending the relevant data from your external system. It makes it easier to test edge cases.

## Edit output data

To edit output data:

1. Run the node to load data.
2. In the **OUTPUT** view, select **JSON** to switch to JSON view.
3. Select **Edit** Image: Edit data icon.
4. Edit your data.
5. Select **Save**. n8n saves your data changes and pins your data.

## Use data from previous executions

You can copy data from nodes in previous workflow executions:

1. Open the left menu.
2. Select **Executions**.
3. Browse the workflow executions list to find the one with the data you want to copy.
4. Select **Open Past Execution** Image: Open past execution icon.
5. Double click the node whose data you want to copy.
6. If it’s table layout, select **JSON** to switch to JSON view.
7. There are two ways to copy the JSON:
  8. Select the JSON you want by highlighting it, like selecting text. Then use **ctrl + c** to copy it.
  9. Select the JSON you want to copy by clicking on a parameter. Then: 1. Hover over the JSON. n8n displays the **Copy** Image: Copy data icon button. 2. Select **Copy** Image: Copy data icon. 3. You can choose what to copy:

- **Copy Item Path** and **Copy Parameter Path** gives you expressions that access parts of the JSON.
  - **Copy Value**: copies the entire selected JSON.
10. Return to the workflow you're working on:
    1. Open the left menu.
    2. Select **Workflows**.
    3. Select **Open**.
    4. Select the workflow you want to open.
  11. Open the node where you want to use the copied data.
  12. If there is no data, run the node to load data.
  13. In the **OUTPUT** view, select **JSON** to switch to JSON view.
  14. Select **Edit** Image: Edit data icon.
  15. Paste in the data from the previous execution.
  16. Select **Save**. n8n saves your data changes and pins your data.
- 

## Data filtering

Search and filter data in the node **INPUT** and **OUTPUT** panels. Use this to check your node's data.

To search:

1. In a node, select **Search** Image: Search icon in the **INPUT** or **OUTPUT** panel.
2. Enter your search term.

n8n filters as you type your search, displaying the objects or rows containing the term.

Filtering is purely visual: n8n doesn't change or delete data. The filter resets when you close and reopen the node.

---

## Data mocking

Data mocking is simulating or faking data. It's useful when developing a workflow. By mocking data, you can:

- Avoid making repeated calls to your data source. This saves time and costs.
- Work with a small, predictable dataset during initial development.
- Avoid the risk of overwriting live data: in the early stages of building your workflow, you don't need to connect your real data source.

## Mocking with real data using data pinning

Using [data pinning](#), you load real data into your workflow, then pin it in the output panel of a node. Using this approach you have realistic data, with only one call to your data source. You can [edit pinned data](#).

Use this approach when you need to configure your workflow to handle the exact data structure and parameters provided by your data source.

-8<- “\_snippets/data/how-to-pin-data.md”

## Generate custom data using the Code or Edit Fields nodes

You can create a custom dataset in your workflow using either the [Code node](#) or the [Edit Fields \(Set\) node](#).

In the Code node, you can create any data set you want, and return it as the node output. In the Edit Fields node, select **Add fields** to add your custom data.

The Edit Fields node is a good choice for small tests. To create more complex datasets, use the Code node.

## Output a sample data set from the Customer Datastore node

The Customer Datastore node provides a fake dataset to work with. Add and execute the node to explore the data.

Use this approach if you need some test data when exploring n8n, and you don't have a real use-case to work with.

---

# Data tables

## Overview

Data tables integrate data storage within your n8n environment. Using data tables, you can save, manage, and interact with data directly inside your workflows without relying on external database systems for scenarios such as:

- Persisting data across workflows in the same project
- Storing markers to prevent duplicate runs or control workflow triggers
- Reusing prompts or messages across workflows
- Storing evaluation data for AI workflows
- Storing data generated from workflow executions
- Combining data from different sources to enrich your datasets
- Creating lookup tables as quick reference points within workflows

## How to use data tables

There are two parts to working with data tables: creating them and interacting with them in workflows.

### Step 1: Creating a data table

1. In your n8n project, select the **Data tables** tab.

2. Click the split button located in the top right corner and select **Create Data table**.

[Image: Data table creation]

3. Enter a descriptive name for your table.

In the table view that appears, you can:

- Add and reorder columns to organize your data
- Add, delete, and update rows
- Edit existing data

## Step 2: Interacting with Data tables in workflows

Interact with data tables in your workflow using the **Data table** node, which allows you to retrieve, update, and manipulate the data stored in a Data table.

See [Data table node](#).

## Considerations and limitations of data tables

- Data tables are suitable for light to moderate data storage. By default, a data table can't contain more than 50MB of data. In self-hosted environments, you can increase this default size limit using the environment variable `N8N_DATA_TABLES_MAX_SIZE_BYTES`.
- When a data table approaches 80% of your storage limit, a warning will alert you. A final warning appears when you reach the storage limit. Exceeding this limit will disable manual additions to tables and cause workflow execution errors during attempts to insert or update data.
- By default, data tables created within a project are accessible to all team members in that project.
- Tables created in a **Personal** space are only accessible by their creator.

## Data tables versus variables

Feature	Data tables	Variables
Unified tabular view	✓	✗
Row-column relationships	✓	✗
Cross-project access	✗	✓
Individual value display	✗	✓
Optimized for short values	✗	✓
Structured data	✓	✗
Scoped to projects	✓	✗
Use values as expressions	✗	✓

## Exporting and importing data

To transfer data between n8n and external tools, use workflows that:

1. Retrieve data from a data table.
2. Export it using an API or file export.
3. Import data into another system or data table accordingly.

[Image: Data export workflow]

---

## Binary data

Binary data is any file-type data, such as image files or documents.

This page collects resources relating to binary data in n8n.

## Working with binary data in your workflows

You can process binary data in n8n workflows. n8n provides nodes to help you work with binary data. You can also use code.

### Nodes

There are three key nodes dedicated to handling binary data files:

- [Read/Write Files from Disk](#) to read and write files from/to the machine where n8n is running.
- [Convert to File](#) to take input data and output it as a file.
- [Extract From File](#) to get data from a binary format and convert it to JSON.

There are separate nodes for working with XML and HTML data:

- [HTML](#)
- [XML](#)

And nodes for performing common tasks:

- [Compression](#)
- [Edit Image](#)
- [FTP](#)

You can trigger a workflow based on changes to a local file using the [Local File trigger](#).

To split or concatenate binary data items, use the [data transformation nodes](#).

### Code

You can use the [Code node](#) to manipulate binary data in your workflows. For example, [Get the binary data buffer](#): get the binary data available in your workflow.

## Configure binary data mode when self-hosting

You can configure how your self-hosted n8n instance handles binary data using the [Binary data environment variables](#). This includes tasks such as setting the storage path and choosing how to store binary data.

Your configuration affects how well n8n scales: [Scaling | Binary data filesystem mode](#).

Reading and writing binary files can have security implications. If you want to disable reading and writing binary data, use the `NODES_EXCLUDE` environment variable. Refer to [Environment variables | Nodes](#) for more information.

---

## Schema Preview

Schema Preview exposes expected schema data from the previous node in the Node Editor without the user having to provide credentials or execute the node. This makes it possible to construct workflows without having to provide credentials in advance. The preview doesn't include mock data, but it does expose the expected fields, making it possible to select and incorporate them into the input of subsequent nodes.

### Using the preview

1. There must be a node with Schema Preview available in your workflow.
  2. When clicking on the details of the next node in the sequence, the Schema Preview data will show up in the Node Editor where schema data would typically be exposed.
  3. Use data from the Schema Preview just as you would other schemas - drag and drop fields as input into your node parameters and settings.
- 

### AI agent

AI agents are artificial intelligence systems capable of responding to requests, making decisions, and performing real-world tasks for users. They use large language models (LLMs) to interpret user input and make decisions about how to best process requests using the information and resources they have available.

### AI chain

AI chains allow you to interact with large language models (LLMs) and other resources in sequences of calls to components. AI chains in n8n don't use persistent memory, so you can't use them to reference previous context (use AI agents for this).

### AI completion

Completions are the responses generated by a model like GPT.

### **AI embedding**

Embeddings are numerical representations of data using vectors. They're used by AI to interpret complex data and relationships by mapping values across many dimensions. Vector databases, or vector stores, are databases designed to store and access embeddings.

### **AI groundedness**

In AI, and specifically in retrieval-augmented generation (RAG) contexts, groundedness and ungroundedness are measures of how much a model's responses accurately reflect source information. The model uses its source documents to generate grounded responses, while ungrounded responses involve speculation or hallucination unsupported by those same sources.

### **AI hallucination**

Hallucination in AI is when an LLM (large language model) mistakenly perceives patterns or objects that don't exist.

### **AI reranking**

Reranking is a technique that refines the order of a list of candidate documents to improve the relevance of search results. Retrieval-Augmented Generation (RAG) and other applications use reranking to prioritize the most relevant information for generation or downstream tasks.

### **AI memory**

In an AI context, memory allows AI tools to persist message context across interactions. This allows you to have a continuing conversations with AI agents, for example, without submitting ongoing context with each message. In n8n, AI agent nodes can use memory, but AI chains can't.

### **AI retrieval-augmented generation (RAG)**

Retrieval-augmented generation, or RAG, is a technique for providing LLMs access to new information from external sources to improve AI responses. RAG systems retrieve relevant documents to ground responses in up-to-date, domain-specific, or proprietary knowledge to supplement their original training data. RAG systems often rely on vector stores to manage and search this external data efficiently.

### **AI tool**

In an AI context, a tool is an add-on resource that the AI can refer to for specific information or functionality when responding to a request. The AI model can use a tool to interact with external systems or complete specific, focused tasks.

### **AI vector store**

A vector store, or vector database, stores mathematical representations of information. Use with embeddings and retrievers to create a database that your AI can access when answering questions.

## API

APIs, or application programming interfaces, offer programmatic access to a service's data and functionality. APIs make it easier for software to interact with external systems. They're often offered as an alternative to traditional user-focused interfaces accessed through web browsers or UI.

### **canvas (n8n)**

The canvas is the main interface for building workflows in n8n's editor UI. You use the canvas to add and connect nodes to compose workflows.

### **cluster node (n8n)**

In n8n, cluster nodes are groups of nodes that work together to provide functionality in a workflow. They consist of a root node and one or more sub nodes that extend the node's functionality.

### **credential (n8n)**

In n8n, credentials store authentication information to connect with specific apps and services. After creating credentials with your authentication information (username and password, API key, OAuth secrets, etc.), you can use the associated app node to interact with the service.

### **data pinning (n8n)**

Data pinning allows you to temporarily freeze the output data of a node during workflow development. This allows you to develop workflows with predictable data without making repeated requests to external services. Production workflows ignore pinned data and request new data on each execution.

### **editor (n8n)**

The n8n editor UI allows you to create and manage workflows. The main area is the canvas, where you can compose workflows by adding, configuring, and connecting nodes. The side and top panels allow you to access other areas of the UI like credentials, templates, variables, executions, and more.

### **entitlement (n8n)**

In n8n, entitlements grant n8n instances access to plan-restricted features for a specific period of time.

Floating entitlements are a pool of entitlements that you can distribute among various n8n instances. You can re-assign a floating entitlement to transfer its access to a different n8n instance.

### **evaluation (n8n)**

In n8n, evaluation allows you to tag and organize execution history and compare it against new executions. You can use this to understand how your workflow performs over time as you make changes. In particular, this is useful while developing AI-centered workflows.

### **expression (n8n)**

In n8n, expressions allow you to populate node parameters dynamically by executing JavaScript code. Instead of providing a static value, you can use the n8n expression syntax to define the value using data from previous nodes, other workflows, or your n8n environment.

### **LangChain**

LangChain is an AI-development framework used to work with large language models (LLMs). LangChain provides a standardized system for working with a wide variety of models and other resources and linking different components together to build complex applications.

### **Large language model (LLM)**

Large language models, or LLMs, are AI machine learning models designed to excel in natural language processing (NLP) tasks. They're built by training on large amounts of data to develop probabilistic models of language and other data.

### **node (n8n)**

In n8n, nodes are individual components that you compose to create workflows. Nodes define when the workflow should run, allow you to fetch, send, and process data, can define flow control logic, and connect with external services.

### **project (n8n)**

n8n projects allow you to separate workflows, variables, and credentials into separate groups for easier management. Projects make it easier for teams to collaborate by sharing and compartmentalizing related resources.

### **root node (n8n)**

Each n8n cluster node contains a single root nodes that defines the main functionality of the cluster. One or more sub nodes attach to the root node to extend its functionality.

### **sub node (n8n)**

n8n cluster nodes consist of one or more sub nodes connected to a root node. Sub nodes extend the functionality of the root node, providing access to specific services or resources or offering specific types of dedicated processing, like calculator functionality, for example.

### **template (n8n)**

n8n templates are pre-built workflows designed by n8n and community members that you can import into your n8n instance. When using templates, you may need to fill in credentials and adjust the configuration to suit your needs.

### **trigger node (n8n)**

A trigger node is a special node responsible for executing the workflow in response to certain conditions. All production workflows need at least one trigger to determine when the workflow should run.

### **workflow (n8n)**

An n8n workflow is a collection of nodes that automate a process. Workflows begin execution when a trigger condition occurs and execute sequentially to achieve complex tasks.

---

## **n8n Cloud**

n8n Cloud is n8n's hosted solution. It provides:

- No technical set up or maintenance for your n8n instance
- Continual uptime monitoring
- Managed OAuth for authentication
- One-click upgrades to the newest n8n versions

[Sign up for n8n Cloud](#)

---

## **Cloud free trial**

When you create a new n8n cloud trial, you have 14 days to try all the features of the [Pro plan](#), including:

- Global variables
- Insights dashboard
- Execution search
- 5 days workflow history to rollback

The trial gives you Pro plan features with limits of 1000 executions and the same computing power as the [Starter plan](#).

## **Upgrade to a paid account**

You can upgrade to a paid n8n account at any time. To upgrade:

1. Log in to your account.
2. Click the **Upgrade** button in the upper-right corner.
3. Select your plan and whether to pay annually or by the month.
4. Select a payment method.

## **Trial expiration**

If you don't upgrade by the end of your trial, the trial will automatically expire and your workspace will be deleted.

## Cancelling your trial

You don't need to cancel your trial. Your trial will automatically expire at the end of the trial period and no charges will occur. All your data will be deleted soon after.

## Enterprise trial

You can contact the sales team if you want to test the [Enterprise plan](#), which includes features such as:

- SSO SAML and LDAP
- Different environments
- External secret store integration
- Log streaming
- Version control using Git

Click the **Contact** button on the [n8n website](#).

---

## Cloud admin dashboard

Instance owners can access the admin dashboard to manage their Cloud instance. This is where you can upgrade your n8n version and set the timezone.

### Access the dashboard from the app

1. [Log in to n8n](#)
2. Select **Admin Dashboard**. n8n opens the dashboard.

### Access the dashboard if the app is offline

If your instance is down, you can still access the admin dashboard. When you log in to the app, n8n will ask you if you want a magic link to access your dashboard. Select **Send magic link**, then check your email for the link.

---

## Update your Cloud version

n8n recommends regularly updating your Cloud version. Check the [Release notes](#) to learn more about changes.

1. [Log in to the n8n Cloud dashboard](#)
2. On your dashboard, select **Manage**.
3. Use the **n8n version** dropdown to select your preferred release version:
  - Latest Stable: recommended for most users.

- Latest Beta: get the newest n8n. This may be unstable.
4. Select **Save Changes** to restart your n8n instance and perform the update.
  5. In the confirmation modal, select **Confirm**.

## Best practices for updating

-8<- “\_snippets/manage-cloud/updating-best-practices.md”

### Automatic update

n8n automatically updates outdated Cloud instances.

If you don't update your instance for 120 days, n8n emails you to warn you to update. After a further 30 days, n8n automatically updates your instance.

---

## Set the Cloud instance timezone

You can change the timezone for your n8n instance. This affects the Schedule Trigger and Date & Time node. Users can configure the timezone for individual workflows in Workflow settings.

1. On your dashboard, select **Manage**.
  2. Change the **Timezone** dropdown to the timezone you want.
- 

## Cloud IP addresses

Outbound traffic may appear to originate from any of:

- 20.79.227.226/32
- 20.113.47.122/32
- 20.218.202.73/32
- 98.67.233.91/32
- 4.182.111.50/32
- 4.182.129.20/32
- 4.182.88.118/32
- 4.182.212.136/32
- 98.67.244.108/32
- 72.144.128.145/32
- 72.144.83.147/32
- 72.144.69.38/32

- 72.144.111.50/32
  - 4.182.128.108/32
  - 4.182.190.144/32
  - 4.182.191.184/32
  - 98.67.233.200/32
  - 20.52.126.0/28
  - 20.218.238.112/28
  - 4.182.64.64/28
  - 20.218.174.0/28
  - 4.184.78.240/28
  - 20.79.32.32/28
  - 51.116.119.64/28
- 

## Cloud data management

There are two concerns when managing data on Cloud:

- Memory usage: complex workflows processing large amounts of data can exceed n8n's memory limits. If this happens, the instance can crash and become inaccessible.
- Data storage: depending on your execution settings and volume, your n8n database can grow in size and run out of storage.

To avoid these issues, n8n recommends that you build your workflows with memory efficiency in mind, and don't save unnecessary data

## Memory limits on each Cloud plan

Current plans:

- Trial: 320MiB RAM, 10 millicore CPU burstable
- Starter: 320MiB RAM, 10 millicore CPU burstable
- Pro-1 (10k executions): 640MiB RAM, 20 millicore CPU burstable
- Pro-2 (50k executions): 1280MiB RAM, 80 millicore CPU burstable
- Enterprise: 4096MiB RAM, 80 millicore CPU burstable

Legacy plans:

- Start: 320MiB RAM, 10 millicore CPU burstable
- Power: 1280MiB RAM, 80 millicore CPU burstable

n8n gives each instance up to 100GB of data storage.

## How to reduce memory consumption in your workflow

The way you build workflows affects how much data they consume when executed. Although these guidelines aren't applicable to all cases, they provide a baseline of best practices to avoid exceeding instance memory.

```
-8<- "_snippets/self-hosting/scaling/reduce-memory-consumption.md"
```

Note that n8n itself consumes memory to run. On average, the software alone uses around 180MiB RAM.

Interactions with the UI also consume memory. Playing around with the workflow UI while it performs heavy executions could also push the memory capacity over the limit.

## How to manage execution data on Cloud

Execution data includes node data, parameters, variables, execution context, and binary data references. It's text-based.

Binary data is non-textual data that n8n can't represent as plain text. This is files and media such as images, documents, audio files, and videos. It's much larger than textual data.

If a workflow consumes a large amounts of data and is past testing stage, it's a good option to stop saving the successful executions.

There are two ways you can control how much execution data n8n stores in the database:

In the admin dashboard:

1. From your workspace or editor, navigate to **Admin Panel**.
2. Select **Manage**.
3. In **Executions to Save** deselect the executions you don't want to log.

In your workflow settings:

1. Select the **Options** Image: Options menu menu.
2. Select **Settings**. n8n opens the **Workflow settings** modal.
3. Change **Save successful production executions** to **Do not save**.

## Cloud data pruning and out of memory incident prevention

### Automatic data pruning

n8n automatically prunes execution logs after a certain time or once you reach the max storage limit, whichever comes first. The pruning always happens from oldest to newest and the limits depend on your Could plan:

- Start and Starter plans: max 2500 executions saved and 7 days execution log retention;
- Pro plans: max 25000 executions saved and 30 days execution log retention;
- Enterprise plan: max 50000 executions saved and unlimited execution log retention time.

## Manual data pruning

Heavier executions and use cases can exceed database capacity despite the automatic pruning practices. In cases like this, n8n will manually prune data to protect instance stability.

1. An alert system warns n8n if an instance is at 85% disk capacity.
2. n8n prunes execution data. n8n does this by running a backup of the instance (workflows, users, credentials and execution data) and restoring it without execution data.

Due to the human steps in this process, the alert system isn't perfect. If warnings are triggered after hours or if data consumption rates are high, there might not be time to prune the data before the remaining disk space fills up.

---

## Change instance ownership

You can change the ownership of an instance by navigating to the **Settings > Personal** page in the owner's account and editing the **Email** field. After making the changes, scroll down and press **Save**. Note that for the change to be effective, the new email address must not be associated with any other n8n account, as each instance can have only one **unique owner email**. If the intended email is already linked to an existing user, that user must be deleted before proceeding.

Changing emails will change the owner of the instance, the email you log in with, and the email your invoices and general communication gets sent to.

If the workspace is deactivated, there will be no **Settings** page and no possibility to change the email address or the owner info.

## Change instance username

It's not currently possible to change usernames.

If you want your instance to have a different name you will need to create a new account and transfer your work into it. [The import/export documentation](#) explains how you can transfer your work to a new n8n instance.

---

## Cloud concurrency

Too many concurrent executions can cause performance degradation and unresponsiveness. To prevent this and improve instance stability, n8n sets concurrency limits for production executions in regular mode.

Any executions beyond the limits queue for later processing. These executions remain in the queue until concurrency capacity frees up, and are then processed in FIFO order.

## Concurrency limits

n8n limits the number of concurrent executions for Cloud instances according to their plan. Refer to [Pricing](#) for details.

You can view the number of active executions and your plan's concurrency limit at the top of a project's or workflow's executions tab.

## Details

Some other details about concurrency to keep in mind:

- Concurrency control applies only to production executions: those started from a webhook or trigger node. It doesn't apply to any other kinds, such as manual executions, sub-workflow executions, or error executions.
- Test evaluations don't count towards concurrency limits. Your test evaluation concurrency limit is equal to, but separate from, your plan's regular concurrency limit.
- You can't retry queued executions. Cancelling or deleting a queued execution also removes it from the queue.
- On instance startup, n8n resumes queued executions up to the concurrency limit and re-enqueues the rest.

## Comparison to queue mode

Concurrency in queue mode is a separate mechanism from concurrency in regular mode. In queue mode, the concurrency settings determine how many jobs each worker can run in parallel. In regular mode, concurrency limits apply to the entire instance.

---

## Download workflows

n8n Cloud instance owners can download workflows from the most recent backup.

You can do this with the [Cloud admin dashboard](#).

## How to download workflows

1. [Log in to n8n](#).
2. Select **Admin Dashboard** to open the dashboard.
3. In the **Manage** section, select the **Export** tab.
4. Select **Download Workflows**.

## Accessing workflows after your free trial

You have **90 days** to download your workflows after your free trial ends. After that, all workflows will be **permanently deleted** and are unrecoverable.

---

# AI Assistant

The n8n AI Assistant helps you build, debug, and optimize your workflows seamlessly. From answering questions about n8n to providing help with coding and expressions, the AI Assistant can streamline your workflow-building process and support you as you navigate n8n's capabilities.

## Current capabilities

The AI Assistant offers a range of tools to support you:

- **Debug helper:** Identify and troubleshoot node execution issues in your workflows to keep them running without issues.
- **Answer n8n questions:** Get instant answers to your n8n-related questions, whether they're about specific features or general functionality.
- **Coding support:** Receive guidance on coding, including SQL and JSON, to optimize your nodes and data processing.
- **Expression assistance:** Learn how to create and refine expressions to get the most out of your workflows.
- **Credential setup tips:** Find out how to set up and manage node credentials securely and efficiently.

## Tips for getting the most out of the Assistant

1. **Engage in a conversation:** The AI Assistant can collaborate with you step-by-step. If a suggestion isn't what you need, let it know! The more context you provide, the better the recommendations will be.
2. **Ask specific questions:** For the best results, ask focused questions (for example, "How do I set up credentials for Google Sheets?"). The assistant works best with clear queries.
3. **Iterate on suggestions:** Don't hesitate to build on the assistant's responses. Try different approaches and keep refining based on the assistant's feedback to get closer to your ideal solution.
4. **Things to try out:**
  - Debug any error you're seeing
  - Ask how to setup credentials
  - "Explain what this workflow does."
  - "I need your help to write code: [Explain your code here]"
  - "How can I build X in n8n?"

## FAQs

### What context does the Assistant have?

The AI Assistant has access to all elements displayed on your n8n screen, excluding actual input and output data values (like customer information). To learn more about what data n8n shares with the

Assistant, refer to [AI in n8n](#).

## Who can use the Assistant?

Any user on a Cloud plan can use the assistant.

## How does the Assistant work?

The underlying logic of the assistant is build with the advanced AI capabilities of n8n. It uses a combination of different [agents](#), specialized in different areas of n8n, RAG to gather knowledge from the docs and the community forum, and custom prompts, [memory](#) and context.

---

# Source control and environments

-8<- “\_snippets/source-control-environments/feature-availability.md”

n8n uses Git-based source control to support environments. Linking your n8n instances to a Git repository lets you create multiple n8n environments, backed by Git branches.

In this section:

- [Understand](#):
  - [Environments in n8n](#): The purpose of environments, and how they work in n8n.
  - [Git and n8n](#): How n8n uses Git.
  - [Branch patterns](#): The possible relationships between n8n instances and Git branches.
- [Set up source control for environments](#): How to connect your n8n instance to Git.
- [Using](#):
  - [Push and pull](#): Send work to Git, and fetch work from Git to your instance.
  - [Copy work between environments](#): How to copy work between different n8n instances.
- [Tutorial: Create environments with source control](#): An end-to-end tutorial, setting up environments using n8n’s recommended configurations.

Related sections:

- [Variables](#): reusable values.
  - [External secrets](#): manage [credentials](#) with an external secrets vault.
- 

# Understand source control and environments

-8<- “\_snippets/source-control-environments/feature-availability.md”

- [Environments in n8n](#): The purpose of environments, and how they work in n8n.

- [Git in n8n](#): How n8n uses Git.
  - [Branch patterns](#): The possible relationships between n8n instances and Git branches.
- 

## Environments in n8n

n8n has built its environments feature on top of Git, a version control software. This document helps you understand:

- The purpose of environments.
- How environments work in n8n.

### Environments: What and why

In software development, the environment is all the infrastructure and tooling around the code, including the tools that run the software, and the specific configuration of those tools. For a more detailed introduction to environments in software development, refer to [Codecademy | Environments](#).

Low-code development in n8n is similar. n8n is where you build and run your workflows. Your instance may have particular configurations: on Cloud, n8n determines the configuration. On self-hosted instances, there are extensive [configuration options](#). You may also have made changes to the settings of your instance. This combination of n8n and your instance's specific configuration and settings is the environment your workflows run in.

There are advantages to having more than one environment. A common pattern is to have different environments for development and production:

- Development: do work and make changes.
- Production: the live environment.

A setup like this helps you make changes to workflows without breaking workflows that are in use.

## Environments in n8n

In n8n, an environment comprises two parts, an n8n instance and a Git branch:

- The n8n instance is where you build and run workflows.
- The Git branch stores copies of the workflows, as well as tags, and variable and credential stubs.

n8n doesn't sync credentials and variable values with Git. You must set up the credentials and variable values manually when setting up a new instance. For more information, refer to [Push and pull | What gets committed](#).

How you copy work between environments depends on your branch and n8n instance configuration:

- Multiple instances, one branch: you can push from one instance to

the Git branch, then pull the work to another instance.

- Multiple instances, multiple branches: you need to create a pull request and merge in your Git provider. For example, if you have development, test, and production branches, each linked to their own instance, you need to merge the development branch into test to make the work from the development instance available on the test instance. Refer to [Copy work between environments](#) for more information, including steps to partially automate the process.

For detailed guidance on pushing and pulling work, refer to [Push and pull](#).

Refer to [Set up source control](#) to learn more about linking your n8n instance to Git, or follow the [Tutorial: Create environments with source control](#) to set up your environments using one of n8n's recommended configurations.

---

## Git and n8n

n8n uses Git to provide source control. To use this feature, it helps to have some knowledge of basic Git concepts. n8n doesn't implement all Git functionality: you shouldn't view n8n's source control as full version control.

This page introduces the Git concepts and terminology used in n8n. It doesn't cover everything you need to set up and manage a repository. The person doing the [Setup](#) should have some familiarity with Git and with their Git hosting provider.

### Git overview

[Git](#) is a tool for managing, tracking, and collaborating on multiple versions of documents. It's the basis for widely used platforms such as [GitHub](#) and [GitLab](#).

### Branches: Multiple copies of a project

Git uses branches to maintain multiple copies of a document alongside each other. Every branch has its own version. A common pattern is to have a main branch, and then everyone who wants to contribute to the project works on their own branch (copy). When they finish their work, their branch is merged back into the main branch.

[Image: Diagram]

### Local and remote: Moving work between your machine and a Git provider

A common pattern when using Git is to install Git on your own computer, and use a Git provider such as GitHub to work with Git in the cloud. In effect, you have a Git repository (project) on GitHub, and work with copies of it on your local machine.

n8n uses this pattern for source control: you'll work with your workflows on your n8n instance, but send them to your Git provider to store them.

## Push, pull, and commit

n8n uses three key Git processes:

- **Push**: send work from your instance to Git. This saves a copy of your workflows and tags, as well as credential and variable stubs, to Git. You can choose which workflows you want to save.
- **Pull**: get the workflows, tags, and variables from Git and load it into n8n. You will need to populate any credentials or variable stubs included in the refreshed items.
- **Commit**: a commit in n8n is a single occurrence of pushing work to Git. In n8n, commit and push happen at the same time.

Refer to [Push and pull](#) for detailed information about how n8n interacts with Git.

---

## Branch patterns

The relationship between n8n instances and Git branches is flexible. You can create different setups depending on your needs.

-8<- “\_snippets/source-control-environments/one-direction.md”

### Multiple instances, multiple branches

This pattern involves having multiple n8n instances, each one linked to its own branch.

You can use this pattern for environments. For example, create two n8n instances, development and production. Link them to their own branches. Push work from your development instance to its branch, do a pull request to move work to the production branch, then pull to the production instance.

-8<- “\_snippets/source-control-environments/multi-instance-multi-branch-pros-cons.md”

[Image: Diagram]

### Multiple instances, one branch

Use this pattern if you want the same workflows, tags, and variables everywhere, but want to use them in different n8n instances.

You can use this pattern for environments. For example, create two n8n instances, development and production. Link them both to the same branch. Push work from development, and pull it into production.

This pattern is also useful when testing a new version of n8n: you can create a new n8n instance with the new version, connect it to the Git branch and test it, while your production instance remains on the older version until you're confident it's safe to upgrade.

-8<- “\_snippets/source-control-environments/multi-instance-one-branch-pros-cons.md”

[Image: Diagram]

## One instance, multiple branches

The instance owner can change which Git branch connects to the instance. The full setup in this case is likely to be a [Multiple instances, multiple branches](#) pattern, but with one instance switching between branches.

This is useful to review work. For example, different users could work on their own instance and push to their own branch. The reviewer could work in a review instance, and switch between branches to load work from different users.

[Image: Diagram]

## One instance, one branch

This is the simplest pattern.

[Image: Diagram]

---

# Set up source control for environments

Link a Git repository to an n8n instance and configure your source control.

n8n uses source control to provide environments. Refer to [Environments in n8n](#) for more information.

## Prerequisites

To use source control with n8n, you need a Git repository with either:

- SSH access (using deploy keys), or
- HTTPS access (using Personal Access Tokens)

This document assumes you are familiar with Git and your Git provider.

## Step 1: Set up your repository and branches

For a new setup:

1. Create a new repository for use with n8n.
2. Create the branches you need. For example, if you plan to have different environments for test and production, set up a branch for each.

To help decide what branches you need for your use case, refer to [Branch patterns](#).

## Step 2: Configure Git in n8n

-8<- “\_snippets/source-control-environments/configure-git-in-n8n.md”

## Step 3: Set up authentication

Configure authentication based on your chosen connection method.

### SSH authentication (using deploy keys)

Set up SSH access by creating a deploy key for the repository using the SSH key from n8n. The key must have write access.

The steps depend on your Git provider. Help links for common providers:

- [GitHub | Managing deploy keys](#)
- [GitLab | Deploy keys](#)

### HTTPS authentication (using Personal Access Tokens)

Create a Personal Access Token (PAT) with repository access permissions.

Help links for creating PATs with common providers:

- [GitHub | Managing personal access tokens](#)
- [GitLab | Personal access tokens](#)
- [Bitbucket | App passwords](#)

Required permissions for your token:

- Repository read/write access
- Contents read/write (for GitHub)
- Source code pull/push (for GitLab)

## Step 4: Connect n8n and configure your instance

1. In **Settings > Environments** in n8n, select **Connect**. n8n connects to your Git repository.
2. Under **Instance settings**, choose which branch you want to use for the current n8n instance.
3. **Optional:** select **Protected instance** to prevent users editing workflows in this instance. This is useful for protecting production

instances.

4. **Optional:** choose a custom color for the instance. This will appear in the menu next to the source control push and pull buttons. It helps users know which instance they're in.
  5. Select **Save settings**.
- 

## Using source control and environments

-8<- “\_snippets/source-control-environments/feature-availability.md”

- Push and pull: Send work to Git, and fetch work from Git to your instance. Understand what gets committed, and how n8n handles merge conflicts.
  - Copy work between environments: How to copy work between different n8n instances.
- 

## Push and pull

If your n8n instance connects to a Git repository, you need to keep your work in sync with Git.

This document assumes some familiarity with Git concepts and terminology. Refer to [Git and n8n](#) for an introduction to how n8n works with Git.

-8<- “\_snippets/source-control-environments/one-direction.md”

## Fetch other people's work

To pull work from Git, select **Pull** Image: Pull icon in the main menu.

-8<- “\_snippets/source-control-environments/push-pull-menu-state.md”

n8n may display a warning about overriding local changes. Select **Pull and override** to override your local work with the content in Git.

When the changes include new variable or credential stubs, n8n notifies you that you need to populate the values for the items before using them.

## Workflow and credential owner may change on pull

When you pull from Git to an n8n instance, n8n tries to assign workflows and credentials to a matching user or project.

If the original owner is a user:

If the same owner is available on both instances (matching email), the owner remains the same. If the original owner isn't on the new instance, n8n sets the user performing the pull as the workflow owner.

If the original owner is a project:

n8n tries to match the original project name to a project name on the new instance. If no matching project exists, n8n creates a new project with the name, assigns the current user as project owner, and imports the workflows and credentials to the project.

## Pulling may cause brief service interruption

If you pull changes to a published workflow, n8n unpublishes the workflow while pulling, then republishes it. This may result in a few seconds of downtime for the workflow.

## Send your work to Git

-8<- “\_snippets/source-control-environments/push.md”

## What gets committed

n8n commits the following to Git:

- Workflows, including their tags and the email address of the workflow owner. You can choose which workflows to push.
- Credential stubs - ID, name and type. Any other fields are included only if they are expressions. You can choose which credentials to push.
- Variable stubs (ID and name)
- Projects
- Folders

## Merge behaviors and conflicts

n8n’s implementation of source control is opinionated. It resolves merge conflicts for credentials and variables automatically. n8n can’t detect conflicts on workflows.

### Workflows

You have to explicitly tell n8n what to do about workflows when pushing or pulling. The Git repository acts as the source of truth.

When pulling, you might get warned that your local copy of a workflow differs from Git, and if you accept, your local copy would be overridden. Be careful not to lose relevant changes when pulling.

When you push, your local workflow will override what’s in Git, so make sure that you have the most up to date version or you risk overriding recent changes.

To prevent the issue described above, you should immediately push your changes to a workflow once you finish working on it. Then it’s safe to pull.

To avoid losing data:

- Design your source control setup so that workflows flow in one

direction. For example, make edits on a development instance, push to Git, then pull to production. Don't make edits on the production instance and push them.

- Don't push all workflows. Select the ones you need.
- Be cautious about manually editing files in the Git repository.

## Credentials, variables and workflow tags

Credentials and variables can't have merge issues, as n8n chooses the version to keep.

On pull:

- If the tag, variable or credential doesn't exist, n8n creates it.
- If the tag, variable or credential already exists, n8n doesn't update it, unless:
  - You set the value of a variable using the API or externally. The new value overwrites any existing value.
  - The credential name has changed. n8n uses the version in Git.
  - The name of a tag has changed. n8n updates the tag name. Be careful when renaming tags as tag names are unique and this could cause database issues when it comes to uniqueness during the pull process.

On push:

- n8n overwrites the entire variables and tags files.
  - If a credential already exists, n8n overwrites it with the changes, but doesn't apply these changes to existing credentials on pull.
- 

## Compare changes with workflow diffs

Workflow diffs allow you to visually compare changes between the workflow you have on an instance and the most recent version saved in your connected Git repository. This helps you understand the exact changes to the workflow before you decide to either push or pull it across different environments.

## Accessing workflow diffs

You can access workflow diffs from two locations:

1. **When pushing changes:** Click the workflow diff icon in the commit modal alongside the workflow you want to review
2. **When pulling changes:** Click the workflow diff icon in the modified changes modal alongside the workflow you want to review

## Understanding the workflow diff view

When you open a workflow diff, n8n displays two workflows stacked vertically:

## When pushing

- **Top panel** (Remote branch): Latest version in your Git repository
- **Bottom panel** (Local): Current locally saved version of the workflow

## When pulling

- **Top panel** (Local): Current version on your n8n instance
- **Bottom panel** (Remote branch): Version you're pulling from the Git repository

In both cases, the top panel always displays the workflow that will update with changes.

The diff view highlights three types of changes:

- **Added nodes and connectors**: New node additions or connectors will show as green along with an “N” icon
- **Modified nodes and connectors**: Modifications to existing nodes or connectors will show as orange along with a “M” icon
- **Deleted nodes and connectors**: Node or connector deletions will show as red along with a “D” icon

## Reviewing node changes

For modified nodes, you can also compare the specific changes. Click modified nodes to show a JSON diff of the changes. You can review the exact configuration for that node before and after the given change.

## Viewing the summary of changes

In the top-right corner, the **changes** button shows the number of changes. This represents the total number of changes across node and node connectors, as well as general workflow settings updates.

## Navigating through each change

You can use the **next** and **previous** arrows in the upper-right corner to cycle through your changes in a logical order. Use the **back** button in the top-left corner to return to the commit or pull modal to select a different workflow to review changes on.

## Who can use workflow diffs

Only users who can push or pull commits for an instance can access workflow diffs:

- instance owners
  - instance admins
  - project admins
-

# Copy work between environments

The steps to send work from one n8n instance to another are different depending on whether you use a single Git branch or multiple branches.

## Single branch

If you have a single Git branch the steps to copy work are:

1. Push work from one instance to the Git branch.
2. Log in to the other instance to pull the work from Git. You can automate pulls.

## Multiple branches

If you have more than one Git branch, you need to merge the branches in your Git provider to copy work between environments. You can't copy work directly between environments in n8n.

A common pattern is:

1. Do work in your developments instance.
2. Push the work to the development branch in Git.
3. Merge your development branch into your production branch.  
Refer to the documentation for your Git provider for guidance on doing this:
  - [GitHub: Creating a pull request](#)
  - [GitLab: Creating merge requests](#)
  - [Git: Basic branching and merging](#)
4. In your production n8n instance, pull the changes. You can automate pulls.

## Automatically send changes to n8n

You can automate parts of the process of copying work, using the /source-control/pull API endpoint. Call the API after merging the changes:

```
curl --request POST \
 --location '<YOUR-INSTANCE-URL>/api/v1/source-control/pull' \
 --header 'Content-Type: application/json' \
 --header 'X-N8N-API-KEY: <YOUR-API-KEY>' \
 --data '{"force": true}'
```

This means you can use a GitHub Action or GitLab CI/CD to automatically pull changes to the production instance on merge.

-8<- “\_snippets/source-control-environments/github-action.md”

---

## Tutorial: Create environments with source control

```
-8<- "_snippets/source-control-environments/feature-availability.md"
```

This tutorial walks through the process of setting up environments end-to-end. You'll create two environments: development and production. It uses GitHub as the Git provider. The process is similar for other providers.

n8n has built its environments feature on top of Git, a version control software. You link an n8n instance to a Git branch, and use a push-pull pattern to move work between environments. You should have some understanding of environments and Git. If you need more information on these topics, refer to:

- [Environments in n8n](#): the purpose of environments, and how they work in n8n.
- [Git and n8n](#): Git concepts and source control in n8n.

## Choose your source control pattern

Before setting up source control and environments, you need to plan your environments, and how they relate to Git branches. n8n supports different [Branch patterns](#). For environments, you need to choose between two patterns: multi-instance, multi-branch, or multi-instance, single-branch. This tutorial covers both patterns.

```
-8<- "_snippets/source-control-environments/one-direction.md"
```

### Multiple instances, multiple branches

[Image: Diagram]

```
-8<- "_snippets/source-control-environments/multi-instance-multi-branch-pros-cons.md"
```

### Multiple instances, one branch

[Image: Diagram]

```
-8<- "_snippets/source-control-environments/multi-instance-one-branch-pros-cons.md"
```

## Set up your repository

Once you've chosen your pattern, you need to set up your GitHub repository.

==== "Multi-branch"

1. [Create a new repository]  
(<https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-new-repository>).
  - \* Make sure the repository is private, unless you want your workflows, tags, and variable and credential stubs exposed to the internet.
  - \* Create the new repository with a README so you can immediately create branches.
1. Create one branch named `production` and another named `development`. Refer to [Creating and deleting branches within your

repository](<https://docs.github.com/en/pull-requests/collaborating-with-pull-requests/proposing-changes-to-your-work-with-pull-requests/creating-and-deleting-branches-within-your-repository>) for guidance.

==== “Single-branch”

[Create a new repository]  
(<https://docs.github.com/en/repositories/creating-and-managing-repositories/creating-a-new-repository>).

- \* Make sure the repository is private, unless you want your workflows, tags, and variable and credential stubs exposed to the internet.
- \* Create the new repository with a README. This creates the `main` branch, which you'll connect to.

## Connect your n8n instances to your repository

Create two n8n instances, one for development, one for production.

### Configure Git in n8n

-8<- “\_snippets/source-control-environments/configure-git-in-n8n.md”

### Set up a deploy key

Set up SSH access by creating a deploy key for the repository using the SSH key from n8n. The key must have write access. Refer to [GitHub | Managing deploy keys](#) for guidance.

### Connect n8n and configure your instance

==== “Multi-branch”

1. In \*\*Settings\*\* > \*\*Environments\*\* in n8n, select \*\*Connect\*\*. n8n connects to your Git repository.
1. Under \*\*Instance settings\*\*, choose which branch you want to use for the current n8n instance. Connect the production branch to the production instance, and the development branch to the development instance.
1. Production instance only: select \*\*Protected instance\*\* to prevent users editing workflows in this instance.
1. Select \*\*Save settings\*\*.

==== “Single-branch”

1. In \*\*Settings\*\* > \*\*Environments\*\* in n8n, select \*\*Connect\*\*.
  1. Under \*\*Instance settings\*\*, select the main branch.
1. Production instance only: select \*\*Protected instance\*\* to prevent users editing workflows in this instance.
1. Select \*\*Save settings\*\*.

## Push work from development

In your development instance, create a few workflows, tags, variables, and credentials.

```
-8<- "_snippets/source-control-environments/push.md"
```

## Pull work to production

Your work is now in GitHub. If you're using a multi-branch setup, it's on the development branch. If you chose the single-branch setup, it's on main.

==== "Multi-branch"

1. In GitHub, create a pull request to merge development into production.
1. Merge the pull request.
1. In your production instance, select \*\*Pull\*\* <span class="n8n-inline-image">[Image: Pull icon]{.off-glb}</span> in the main menu.

==== "Single-branch"

In your production instance, select \*\*Pull\*\* <span class="n8n-inline-image">[Image: Pull icon]{.off-glb}</span> in the main menu.

```
-8<- "_snippets/source-control-environments/push-pull-menu-state.md"
```

## Optional: Use a GitHub Action to automate pulls

If you want to avoid logging in to your production instance to pull, you can use a [GitHub Action](#) and the [n8n API](#) to automatically pull every time you push new work to your production or main branch.

```
-8<- "_snippets/source-control-environments/github-action.md"
```

## Next steps

Learn more about:

- [Environments in n8n](#) and [Git and n8n](#)
  - [Source control patterns](#)
- 

## External secrets

You can use an external secrets store to manage [credentials](#) for n8n.

n8n stores all credentials encrypted in its database, and restricts access to them by default. With the external secrets feature, you can store sensitive credential information in an external vault, and have n8n load it in when required. This provides an extra layer of security and allows you to manage credentials used across multiple [n8n environments](#) in one central place.

## Connect n8n to your secrets store

1. In n8n, go to **Settings > External Secrets**.
2. Select **Set Up** for your store provider.
3. Enter the credentials for your provider:
  - Azure Key Vault: Provide your **vault name**, **tenant ID**, **client ID**, and **client secret**. Refer to the Azure documentation to [register a Microsoft Entra ID app and create a service principal](#). n8n supports only single-line values for secrets.
  - AWS Secrets Manager: provide your **access key ID**, **secret access key**, and **region**. The IAM user must have the `secretsmanager>ListSecrets`, `secretsmanager>BatchGetSecretValue`, and `secretsmanager>GetSecretValue` permissions.

To give n8n access to all secrets in your AWS Secrets Manager, you can attach the following policy to the IAM user:

```
json
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "AccessAllSecrets",
 "Effect": "Allow",
 "Action": [
 "secretsmanager>ListSecrets",
 "secretsmanager>BatchGetSecretValue",
 "secretsmanager>GetResourcePolicy",
 "secretsmanager>GetSecretValue",
 "secretsmanager>DescribeSecret",
 "secretsmanager>ListSecretVersionIds",
],
 "Resource": "*"
 }
]
}
```

You can also be more restrictive and give n8n access to select specific AWS Secret Manager secrets. You still need to allow the `secretsmanager>ListSecrets` and `secretsmanager>BatchGetSecretValue` permissions to access all resources. These permissions allow n8n to retrieve ARN-scoped secrets, but don't provide access to the secret values.

Next, you need set the scope for the `secretsmanager>GetSecretValue` permission to the specific Amazon Resource Names (ARNs) for the secrets you wish to share with n8n. Ensure you use the correct region and account ID in each resource ARNs. You can find the ARN details in the AWS dashboard for your secrets.

For example, the following IAM policy only allows access to secrets with a name starting with `n8n` in your specified AWS account and region:

```
{
 "Version": "2012-10-17",
 "Statement": [
 {
 "Sid": "ListingSecrets",
 "Effect": "Allow",
 "Action": [
 "secretsmanager>ListSecrets",
 "secretsmanager>BatchGetSecretValue"
],
 "Resource": "*"
 },
 {
 "Sid": "RetrievingSecrets",
 "Effect": "Allow",
 "Action": [
 "secretsmanager>GetSecretValue"
],
 "Resource": "arn:aws:secretsmanager:us-east-1:123456789012:secret:n8n-12345678-1234-1234-1234-1234567890ab"
 }
]
}
```

```

 "Action": [
 "secretsmanager:GetSecretValue",
 "secretsmanager:DescribeSecret"
],
 "Resource": [
 "arn:aws:secretsmanager:us-west-
2:123456789000:secret:n8n*"
]
}
]
}

```

For more IAM permission policy examples, consult the [AWS documentation](#).

- HashiCorp Vault: provide the **Vault URL** for your vault instance, and select your **Authentication Method**. Enter your authentication details. Optionally provide a namespace.
  - Refer to the HashiCorp documentation for your authentication method: [Token auth method](#)  
[AppRole auth method](#)  
[Userpass auth method](#)
  - If you use vault namespaces, you can enter the namespace n8n should connect to. Refer to [Vault Enterprise namespaces](#) for more information on HashiCorp Vault namespaces.
- Infisical: provide a **Service Token**. Refer to Infisical’s [Service token](#) documentation for information on getting your token. If you self-host Infisical, enter the **Site URL**.
- Google Cloud Platform: provide a **Service Account Key** (JSON) for a service account that has at least these roles: Secret Manager Secret Accessor and Secret Manager Secret Viewer. Refer to Google’s [service account documentation](#) for more information.

4. **Save** your configuration.
5. Enable the provider using the **Disabled / Enabled** toggle.

## Use secrets in n8n credentials

To use a secret from your store in an n8n credential:

1. Create a new credential, or open an existing one.
2. On the field where you want to use a secret:
  1. Hover over the field.
  2. Select **Expression**.
3. In the field where you want to use a secret, enter an [expression](#) referencing the secret name: js {{ \$secrets.<vault-name>. <secret-name> }} <vault-name> is either vault (for HashiCorp) or infisical or awsSecretsManager. Replace <secret-name> with the name as it appears in your vault.

## Using external secrets with n8n environments

n8n's [Source control and environments](#) feature allows you to create different n8n environments, backed by Git. The feature doesn't support using different credentials in different instances. You can use an external secrets vault to provide different credentials for different environments by connecting each n8n instance to a different vault or project environment.

For example, you have two n8n instances, one for development and one for production. You use Infisical for your vault. In Infisical, create a project with two environments, development and production. Generate a token for each Infisical environment. Use the token for the development environment to connect your development n8n instance, and the token for your production environment to connect your production n8n instance.

## Using external secrets in projects

To use external secrets in an [RBAC project](#), you must have an [instance owner](#) or [instance admin](#) as a member of the project.

## Troubleshooting

### Infisical version changes

Infisical version upgrades can introduce problems connecting to n8n. If your Infisical connection stops working, check if there was a recent version change. If so, report the issue to [help@n8n.io](mailto:help@n8n.io).

### Only set external secrets on credentials owned by an instance owner or admin

Due to the permissions that instance owners and admins have, it's possible for owners and admins to update credentials owned by another user with a secrets expression. This will appear to work in preview for an instance owner or admin, but the secret won't resolve when the workflow runs in production.

Only use external secrets for credentials that are owned by an instance admin or owner. This ensures they resolve correctly in production.

---

## Log streaming

Log streaming allows you to send events from n8n to your own logging tools. This allows you to manage your n8n monitoring in your own alerting and logging processes.

## Set up log streaming

To use log streaming, you have to add a streaming destination.

1. Navigate to **Settings > Log Streaming**.

2. Select **Add new destination**.
3. Choose your destination type. n8n opens the **New Event Destination** modal.
4. In the **New Event Destination** modal, enter the configuration information for your event destination. These depend on the type of destination you're using.
5. Select **Events** to choose which events to stream.
6. Select **Save**.

## Events

The following events are available. You can choose which events to stream in **Settings > Log Streaming > Events**.

- Workflow
  - Started
  - Success
  - Failed
- Node executions
  - Started
  - Finished
- Audit
  - User signed up
  - User updated
  - User deleted
  - User invited
  - User invitation accepted
  - User re-invited
  - User email failed
  - User reset requested
  - User reset
  - User credentials created
  - User credentials shared
  - User credentials updated
  - User credentials deleted
  - User API created
  - User API deleted
  - Package installed
  - Package updated
  - Package deleted
  - Workflow created
  - Workflow deleted
  - Workflow updated
- AI node logs
  - Memory get messages
  - Memory added message
  - Output parser get instructions
  - Output parser parsed
  - Retriever get relevant documents
  - Embeddings embedded document
  - Embeddings embedded query
  - Document processed
  - Text splitter split
  - Tool called
  - Vector store searched
  - LLM generated
  - Vector store populated
- Runner
  - Task requested

- Response received
- Queue
  - Job enqueued
  - Job dequeued
  - Job completed
  - Job failed
  - Job stalled

## Destinations

n8n supports three destination types:

- A syslog server
  - A generic webhook
  - A Sentry client
- 

## Insights

Insights gives instance owners and admins visibility into how workflows perform over time. This feature consists of three parts:

- **Insights summary banner**: Shows key metrics about your instance from the last 7 days at the top of the overview space.
- **Insights dashboard**: A more detailed visual breakdown with per-workflow metrics and historical comparisons.
- **Time saved (Workflow ROI)**: For each workflow, you can choose to set a fixed amount of time saved per workflow, or dynamically calculate time saved based on the execution path taken on a specific workflow.

### Insights summary banner

n8n collects several metrics for both the insights summary banner and dashboard. They include:

- Total production executions (not including sub-workflow executions or manual executions)
- Total failed production executions
- Production execution failure rate
- Time saved (when set on at least one or more active workflows)
- Run time average (including wait time from any wait nodes)

### Insights dashboard

Those on the Pro and Enterprise plans can access the **Insights** section from the side navigation. Each metric from the summary banner is also clickable, taking you to the corresponding chart.

The insights dashboard also has a table showing individual insights from each workflow including total production executions, failed production executions, failure rate, time saved, and run time average.

### Insights time periods

By default, the insights summary banner and dashboard show a rolling 7 day window with a comparison to the previous period to identify increases or decreases for each metric. On the dashboard, paid plans also display data for other date ranges:

- Pro: 7 and 14 days
- Enterprise: 24 hours, 7 days, 14 days, 30 days, 90 days, 6 months, 1 year

## Setting the time saved by a workflow

For each workflow, you can track how much time it saves you. This setting helps you calculate how much time automating a process saves over time vs the manual effort to complete the same task or process.

Once configured, n8n calculates the amount of time the workflow saves you based on the number of production executions and displays it on the summary banner and insights dashboard.

You can choose between two methods for calculating time saved:

### Fixed time saved

With fixed time saved, you set a single time value that applies to every production execution of the workflow, regardless of which path the execution takes.

To configure fixed time saved:

1. Navigate to the workflow
2. Select the three dots menu in the top right and select **Settings**
3. In the **Estimated time saved** dropdown, select **Fixed**
4. Enter the number of minutes of work each execution saves
5. Save your settings

### Dynamic time saved

Dynamic time saved calculates time savings based on the actual execution path taken, accounting for workflows where different execution paths save different amounts of time.

To configure dynamic time saved:

1. Navigate to the workflow
2. Select the three dots menu in the top right and select **Settings**
3. In the **Estimated time saved** dropdown, select **Dynamic**
4. Save your settings
5. Add **Time Saved** nodes to your workflow at the points where time is saved
6. For each Time Saved node, configure:
  - **Time saved**: The amount of time saved (in minutes)
  - **Calculation mode**: Choose whether to calculate the time saved once for all items in an execution, or per item which will multiply minutes saved by the total number of input items

When you use dynamic time saved, n8n adds up the time from all Time Saved nodes that execute during a workflow run to calculate the total time saved for that execution.

## Disable or configure insights metrics collection

If you self-host n8n, you can disable or configure insights and metrics collection using [environment variables](#).

## Insights FAQs

### Which executions do n8n use to calculate the values in the insights banner and dashboard?

n8n insights only collects data from production executions (for example, those from active workflows triggered on a schedule or a webhook) from the main (parent) workflow. This means that it doesn't count manual (test) executions or executions from sub-workflows or error workflows.

### Does n8n use historic execution data when upgrading to a version with insights?

n8n only starts collecting data for insights once you update to the first supported version (1.89.0). This means it only reports on executions from that point forward and you won't see execution data in insights from prior periods.

---

## License Key

To enable certain licensed features, you must first activate your license. You can do this either through the UI or by setting environment variables.

### Add a license key using the UI

In your n8n instance:

1. Log in as **Admin** or **Owner**.
2. Select **Settings > Usage and plan**.
3. Select **Enter activation key**.
4. Paste in your license key.
5. Select **Activate**.

### Add a license key using an environment variables

In your n8n configuration, set `N8N_LICENSE_ACTIVATION_KEY` to your license key. If the instance already has an activated license, this variable will have no effect.

Refer to [Environment variables](#) to learn more about configuring n8n.

## Allowlist the license server IP addresses

n8n uses Cloudflare to host the license server. As the specific IP addresses can change, you need to allowlist the [full range of Cloudflare IP addresses](#) to ensure n8n can always reach the license server.

---

## Release notes

New features and bug fixes for n8n.

You can also view the [Releases](#) in the GitHub repository.

-8<- “\_snippets/self-hosting/installation/latest-next-version.md”  
-8<- “\_snippets/update-n8n.md”

## Semantic versioning in n8n

n8n uses [semantic versioning](#). All version numbers are in the format MAJOR.MINOR.PATCH. Version numbers increment as follows:

- MAJOR version when making incompatible changes which can require user action.
- MINOR version when adding functionality in a backward-compatible manner.
- PATCH version when making backward-compatible bug fixes.

## n8n@2.0.0

View the [commits](#) for this version.

**Release date:** 2025-12-05

### Major Version Change

n8n 2.0.0 is a hardening release, not a feature release. It strengthens n8n’s position as an enterprise-grade platform with secure-by-default execution, removal of legacy options that caused edge-case bugs, and better performance under load. The goal is a more predictable foundation you can rely on for mission-critical workflows.

This release is currently in **beta**. There’s no urgency to upgrade immediately — take time to review the breaking changes and assess your workflows using the migration tool before upgrading.

For the full story behind 2.0, read our [announcement blog post](#).

### Breaking changes

Version 2.0 includes breaking changes across security defaults, data handling, and configuration. Key changes include:

- Task runners enabled by default (Code node executions now run in isolated environments)

- Environment variable access blocked from Code nodes by default
- ExecuteCommand and LocalFileTrigger nodes disabled by default
- In-memory binary data mode removed

Review the complete list and migration guidance in the [v2.0 breaking changes docs](#).

## Before you upgrade

Use the **Migration Report** tool to identify workflow-level and instance-level issues that need attention before upgrading.

See the [v2.0 migration tool docs](#) for details.

## Product updates

### Publish / Save workflow paradigm

n8n 2.0 introduces a safer approach to updating live workflows. The Save button now preserves your edits without changing production. A new Publish button lets you explicitly push changes live when ready. See [Publish workflows](#) for details.

### Canvas and navigation improvements

Subtle refinements to the workflow editor canvas and reorganized sidebar navigation.

For full release details, refer to [Releases](#) on GitHub.

---

## Release notes pre 2.0

New features and bug fixes for n8n.

You can also view the [Releases](#) in the GitHub repository.

-8<- “\_snippets/self-hosting/installation/latest-next-version.md”  
-8<- “\_snippets/update-n8n.md”

## Semantic versioning in n8n

n8n uses [semantic versioning](#). All version numbers are in the format MAJOR.MINOR.PATCH. Version numbers increment as follows:

- MAJOR version when making incompatible changes which can require user action.
- MINOR version when adding functionality in a backward-compatible manner.
- PATCH version when making backward-compatible bug fixes.

## n8n@2.0.0-rc.4

View the [commits](#) for this version.

**Release date:** 2025-12-05

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.123.3

View the [commits](#) for this version.

**Release date:** 2025-12-05

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@2.0.0-rc.3

View the [commits](#) for this version.

**Release date:** 2025-12-04

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@2.0.0-rc.1

View the [commits](#) for this version.

**Release date:** 2025-12-04

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.123.2

View the [commits](#) for this version.

**Release date:** 2025-12-04

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@2.0.0-rc.2

View the [commits](#) for this version.

**Release date:** 2025-12-04

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.123.1

View the [commits](#) for this version.

**Release date:** 2025-12-03

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@2.0.0-rc.0

View the [commits](#) for this version.

**Release date:** 2025-12-02

This release contains bug fixes.

### Contributors

[farzad528](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.123.1

View the [commits](#) for this version.

**Release date:** 2025-12-03

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.123.0

View the [commits](#) for this version.

**Release date:** 2025-12-01

This release contains bug fixes.

### Contributors

[farzad528](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.122.5

View the [commits](#) for this version.

**Release date:** 2025-12-04

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.122.4

View the [commits](#) for this version.

**Release date:** 2025-11-28

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.122.3

View the [commits](#) for this version.

**Release date:** 2025-11-26

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.122.2

View the [commits](#) for this version.

**Release date:** 2025-11-25

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.122.0

View the [commits](#) for this version.

**Release date:** 2025-11-24

This release contains bug fixes and features.

### MCP Client node

The **MCP Client node** allows you to connect to and use tools exposed by remote MCP servers directly within your workflow.

Previously, this functionality was only available through AI Agents using the **MCP Client Tool**. With this standalone node, you can now call MCP servers *from any step* in your workflow without relying on an Agent - providing greater flexibility and control over how and where you integrate external tools.

### Custom project roles

#### What's new

You can now define **custom project roles** that precisely control what a user can do **inside a project**.

- Three **system roles** remain built-in and non-editable (Admin, Editor, Viewer)
- You can **create your own roles** (e.g. "Workflow Builder", "Credential Manager", "Read-only Analyst") with granular permissions per resource:
  - Project
  - Folders
  - Workflows
  - Credentials
  - Data tables / Variables
  - Pushing changes via source control

Example: a “Workflow Builder” role that can:

- View everything in the project
- Create and edit workflows and folders
- Use existing credentials but not create or edit credentials

[Image: Project roles]

Project roles is now available in the settings of your n8n Enterprise instance.

### How it works

1. Go to the new Project roles section (instance admins only).
2. Start from a preset (Admin, Editor, Viewer) or from scratch.
3. Configure CRUD-style permissions.
4. Save the role. It now becomes available in the project member picker alongside system roles.
5. Assign these roles to project members.

Custom project roles let you assign permissions that match your organization’s needs, improving governance and security in a clear, transparent, and manageable way.

Custom project roles is only available on **n8n enterprise**.

### User provisioning via SSO

If you’re using SAML or OIDC, your Identity Provider (Okta, Azure AD, Google Workspace, etc.) can now automatically assign each user’s instance role and project roles when they log in.

When enabled, administrators can map a group or individual on their SSO identity provider to an instance role and one or more project roles.

This allows you to manage access centrally from your IdP with automated onboarding and off-boarding, eliminating manual access updates inside the n8n UI

Learn more in our dedicated documentation pages.

For [OIDC](#) For [SAML](#)

Notes: User provisioning is not real-time and updates apply on the user’s next login.

User provisioning is only available on n8n Enterprise.

- The AI Workflow Builder now uses improved internal best-practice instructions, which helps it generate higher quality workflows for users.
- Core updates allow creating data tables from CSV files, configurable workflow statistics, and improved user role provisioning for enterprise, including SSO settings integration.
- Enterprise customers using Single Sign On via SAML or OIDC can now provision the instance role and project role of each user in their organisation from their Identity Provider.
  - See the [new SAML documentation](#)
  - See the [new OIDC documentation](#)
- The editor features a dismissable callout for scaling mode, CSV download for data tables, custom project roles (Beta), and enhanced data table visibility.

- Additional improvements include Stripe billing meter events, and updates to binary data handling and workflow search.

## Contributors

[mi1os](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.121.3

View the [commits](#) for this version.

**Release date:** 2025-11-26

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.121.2

View the [commits](#) for this version.

- Fixed an issue with MCP access scope in the core system.
- Updated editor to improve notice background colors for better visibility.

This release also contains bug fixes. **Release date:** 2025-11-20

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.121.1

View the [commits](#) for this version.

- Prepared patch release 1.121.1 by updating version numbers across all packages.
- Added a changelog entry for version 1.121.1 to document the release.

**Release date:** 2025-11-19

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.121.0

View the [commits](#) for this version.

**Release date:** 2025-11-18

### Instance-level MCP connections (beta)

You can now enable MCP connections at the instance level, giving MCP-compatible AI platforms access to all selected workflows through a single oAuth-secured connection.

Once connected, these platforms can discover and interact with your workflows directly from their own interfaces, so you can observe, query, and trigger n8n workflows without switching contexts. As you add new MCP-enabled workflows to your instance, they automatically become available through the same connection without additional setup.

❖ How to: To enable workflows through an instance-level MCP connection, both your instance and each workflow must be opted in. - To turn this feature on in your n8n instance, go to the settings page for the instance and switch the Enable MCP Access toggle to on. - To allow access to individual workflows, go to the settings in each workflow and switch the Available in MCP toggle to on.

❖ Keep in mind:

- To keep access organized and auditable, consider using a workspace or team account when connecting on behalf of an organization. - All platforms connected to your instance MCP will have access to each of the workflows you have enabled. At this time, you cannot limit access of individual workflows to particular platforms.

This feature simplifies integration and improves visibility across AI platforms that support MCP, helping you use your n8n workflows directly in the tools where you already work and experiment. [Learn more in documentation](#).

[Image: Instance MCP]

Enable MCP for an instance in instance settings.

[Image: Instance MCP]

Enable MCP for each workflow in workflow settings.

## Contributors

[vicalca](#)

[datpp-ct](#)

[cesars-gh](#)

[farzad528](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.120.4

View the [commits](#) for this version.

**Release date:** 2025-11-19

- Prepared for release 1.120.4 by updating version numbers across all core, backend, AI/LLM, frontend, and tooling packages.
- Added a changelog entry for version 1.120.4 to document the release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.120.2

View the [commits](#) for this version.

**Release date:** 2025-11-14

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.120.1

View the [commits](#) for this version.

**Release date:** 2025-11-11

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.120.0

View the [commits](#) for this version.

**Release date:** 2025-11-10

This release contains **security updates** and bug fixes. We recommend all users upgrade promptly to version 1.120 or higher.

## Contributors

[rothenbergt](#)

## n8n@1.119.2

View the [commits](#) for this version.

**Release date:** 2025-11-14

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.119.1

View the [commits](#) for this version.

**Release date:** 2025-11-10

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.118.2

View the [commits](#) for this version.

**Release date:** 2025-11-05

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.119.0

View the [commits](#) for this version.

**Release date:** 2025-11-03

This release includes multiple bug fixes for AI Agent, task runners, editor, and integrations, as well as new features like improved workflow settings, AWS Assume Role credentials, and enhanced security and audit capabilities.

### Guardrails Node

The Guardrails node provides a set of rules and policies that control an AI agent's behavior by filtering its inputs and outputs. This helps safeguard from malicious input and from generating unsafe or undesirable responses. There are two operations:

- Check Text for Violations: Validate text against a set of policies (e.g. NSFW, prompt injection).
- Sanitize Text: Detects and replaces specific data such as PII, URLs, or secrets with placeholders.

The default presets and prompts are adapted from the open-source [guardrails package](#) made available by OpenAI.

For more info, see [Guardrails documentation](#)

### OAuth for MCP Client Tool node

The MCP Client Tool now supports OAuth authentication, enabling secure connections to APIs that implement delegated access flows. [Learn more](#) about the MCP Client Tool node.

[Image: Authentication options in MCP Client Tool node]  
Authentication options in MCP Client Tool node

### Contributors

[cesars-gh](#)  
[ongdisheng](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.118.1

View the [commits](#) for this version.

**Release date:** 2025-10-28

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.117.3

View the [commits](#) for this version.

**Release date:** 2025-10-28

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.118.0

View the [commits](#) for this version.

**Release date:** 2025-10-27

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## Project-level variables

You can now create variables at either the global or project level.

- Project-level variables are only visible and usable by members of that project.
- Global variables can be overridden at the project level for more specific configurations.
- The maximum character limit has been increased to 1,000 to support longer values.

This update improves security and flexibility as part of our ongoing effort to isolate data and settings per project.

## Improvements to Environments

We've made several updates to improve reliability, consistency, and control across environments.

- **Projects are now fully synchronized across environments.** When a workflow is pushed within a project, it's automatically created and synced in all environments where it's pulled. Moving, renaming, or deleting projects is now reflected everywhere, keeping instances aligned.
- **Improved protection for read-only environments.** Read-only instances now block all changes, including folder moves and workflow ownership updates, ensuring production environments stay stable and isolated.
- **Bug fix: tag mapping synchronization.** Fixed an issue where tag mappings only updated when tags changed. They now sync on every commit for consistent tagging across environments.

## Custom date ranges for Insights

You can now define custom date ranges instead of using fixed ones, giving you more control over the data you analyze.

## n8n@1.117.2

View the [commits](#) for this version.

**Release date:** 2025-10-27

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.117.1

View the [commits](#) for this version.

**Release date:** 2025-10-24

This release contains bug fixes.

### AI Workflow Builder is now available to Enterprise Cloud users.

AI Workflow Builder turns prompts into workflows. Describe what you want to build, and n8n will generate a draft workflow by adding, configuring, and connecting nodes for you. From there, you can refine and expand the workflow directly in the editor.

#### What's new:

- Previously available to Starter and Pro users, AI Workflow Builder is now accessible to Enterprise Cloud users as well, with 1,000 monthly credits.
- Supported on n8n version 1.115+. **If you don't see the feature yet, open /settings/usage to trigger a license refresh.**
- We've fixed a bug and now cloud users on v1.117.1 onwards will have access to a more reliable builder.
- We're currently working on bringing AI Workflow Builder to self-hosted users as well, including Community, Business, and Enterprise.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.117.0

View the [commits](#) for this version.

**Release date:** 2025-10-21

This release contains bug fixes.

## Contributors

[jackfrancismurphy](#)

[JiriDeJonghe](#)

[ramkrishna2910](#)

Oracle and/or its affiliates ([sudarshan12s](#))

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.116.2

View the [commits](#) for this version.

**Release date:** 2025-10-21

This release contains a bug fix.

## n8n@1.115.4

View the [commits](#) for this version.

**Release date:** 2025-10-21

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.116.1

View the [commits](#) for this version.

**Release date:** 2025-10-14

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.116.0

View the [commits](#) for this version.

**Release date:** 2025-10-13

This release contains bug fixes.

### Data migration tool

You can now easily migrate n8n data between different database types. This new tooling currently supports SQLite and Postgres, making the transition to a scaling database choice simpler, allowing you to take your data with you.

The tooling comes in the form of two new CLI commands, `export:entities` and `import:entities`

**Export** The new export command lets you export data from your existing n8n database (SQLite / Postgres), producing a set of encrypted files within a compressed directory for you to move around and use with the import commands.

For details, see [Export entities](#)

**Import** The new import command allows you to read from a compressed and encrypted set of files generated from the new export command, and import them in to your new database of choice (SQLite / Postgres) to be used with your n8n instance.

For details, see [Import entities](#)

## Contributors

[JHTosas](#)  
[clesecq](#)  
[Gulianrdgd](#)  
[tishun](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.116.0

View the [commits](#) for this version.

**Release date:** 2025-10-13

This release contains bug fixes.

## Contributors

[JHTosas](#)  
[clesecq](#)  
[Gulianrdgd](#)  
[tishun](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.115.3

View the [commits](#) for this version.

**Release date:** 2025-10-14

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.115.2

View the [commits](#) for this version.

**Release date:** 2025-10-10

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.114.4

View the [commits](#) for this version.

**Release date:** 2025-10-07

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.115.0

View the [commits](#) for this version.

**Release date:** 2025-10-06

This release contains bug fixes.

## AI Workflow Builder (Beta)

**AI Workflow Builder** turns your natural language prompts into working automations. Describe what you want to build, and n8n will generate a draft workflow by adding and configuring nodes and wiring up the logic for you. From there, you can refine, expand, or adjust the workflow directly in the editor.

This feature helps you move from idea to implementation faster and without losing technical control. It's especially helpful when starting from a blank canvas, validating an approach, or exploring new nodes and capabilities. Multi-turn interaction lets you iterate in conversation, turning your ideas into structured, production-ready workflows step by step.

Learn more about how we're building this feature in our [forum post](#).

### Availability:

- This feature is initially going to be available for Cloud users on the 14-day Trial, Starter and Pro plans.
- Availability for Enterprise users on Cloud will follow in a future update.
- We are actively exploring the best way to bring this feature to self-hosted users.

### Rollout timing:

- To ensure the smoothest experience for all users, this feature will be rolled out to users on version 1.115.0 over the course of a week so you may not have access to the feature immediately when you upgrade to 1.115.0.

**Credit limits by plan:** This feature will have monthly credit limits [by plan](#).

- Each prompt/interaction with the AI Workflow Builder consumes one credit.
- Trial users have access to 20 credits, Starter plans have 50 per month and Pro plans will have 150 credits per month.
- At this time, there will not be a way to access additional credits within your plan, however we are exploring this.

Learn more about AI Workflow Builder in [documentation](#).

## Source Control: Added HTTPS support

You can now connect to Git repositories via HTTPS in addition to SSH, making Source Control usable in environments where SSH is restricted.

[Image: Source Control]  
HTTPS is now supported as a connection type in Environments.

## Contributors

[baileympearson](#)  
[h40huynh](#)  
[Ankit-69k](#)  
[francisfontoura](#)  
[iocanel](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.114.3

View the [commits](#) for this version.

**Release date:** 2025-10-06

This release contains bug fixes.

## n8n@1.114.2

View the [commits](#) for this version.

**Release date:** 2025-10-02

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.114.1

View the [commits](#) for this version.

**Release date:** 2025-10-02

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.114.0

View the [commits](#) for this version.

**Release date:** 2025-09-29

This release contains core updates, editor improvements, project updates, performance improvements, and bug fixes.

## Contributors

[nealzhu3](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.113.3

View the [commits](#) for this version.

**Release date:** 2025-09-26

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.112.6

View the [commits](#) for this version.

**Release date:** 2025-09-26

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.113.2

View the [commits](#) for this version.

**Release date:** 2025-09-24

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## Python task runner

This version introduces the **Python task runner** as a beta feature. This feature secures n8n's Python sandbox and enables users to run real Python modules in n8n workflows. The original Pyodide-based implementation will be phased out.

This is a **breaking change** that replaces Pyodide - see [here](#) for a list of differences. Any Code node set to the legacy python parameter will need to be manually updated to use the new pythonNative parameter. Any Code node script set to python and relying on Pyodide syntax is likely to need to be manually adjusted to account for breaking changes.

- For self-hosting users, see [here](#) for deployment instructions for task runners going forward and how to install extra dependencies.
- On n8n Cloud, this will be a gradual transition. If in your n8n Cloud instance the Code node offers an option named “Python (Native) (Beta)”, then your instance has been transitioned to native Python and you will need to look out for any breaking changes.  
Imports are disabled for security reasons at this time.

The native Python runner is currently in beta and is subject to change as we find a balance between security and usability. Your feedback is welcome.

## n8n@1.112.5

View the [commits](#) for this version.

**Release date:** 2025-09-24

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.113.1

View the [commits](#) for this version.

**Release date:** 2025-09-23

This release contains bug fixes.

### Data tables

We're excited to introduce **data tables**, bringing built-in data storage to n8n. You can now store and query structured data directly inside the platform, without relying on external databases for many common automation scenarios. Track workflow state between runs, store tokens or session data, keep product or customer reference tables, or stage intermediate results for multi-step processes.

Previously, persisting data meant provisioning and connecting to an external store such as Redis or Google Sheets. That added credential setup, infrastructure overhead, latency, and constant context switching. **Data tables** eliminate that friction and keeps your data easily editable and close to your workflows.

Data tables are available today on all plans. They currently support numbers, strings, and datetimes with JSON support coming soon. On Cloud, each instance can store up to 50 MB. On self-hosted setups, the default is also 50 MB, but this limit can be adjusted if your infrastructure allows.

[\[Image: Data tables\]](#)  
[Overview of data tables](#)

#### \* How to:

##### Create a data table

- From the canvas, open the **Create workflow** dropdown and select **Create Data table**.
- Or, go to the **Overview** panel on the left-side navigation bar and open the **Data tables** tab.

##### Use a data table in your workflow

- Add the **Data table node** to your workflow to get, update, insert, upsert, or delete rows.

##### Adjust the storage limit (self-hosted only)

- Change the default 50 MB limit with the environment variable: `N8N_DATA_TABLES_MAX_SIZE_BYTES`. [See configuration docs](#).

#### Keep in mind

- Data tables don't currently support foreign keys or default values.
- For now, all data tables are accessible to everyone in a project.  
More granular permissions and sharing options are planned.

Learn more about [data tables](#) and the [Data table node](#).

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.112.4

View the [commits](#) for this version.

**Release date:** 2025-09-23

This release contains an editor improvement.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.113.0

View the [commits](#) for this version.

**Release date:** 2025-09-22

This release contains core updates, editor improvements, a new node, node updates, and bug fixes.

### SSO improvements

We've made updates to strengthen Single Sign-On (SSO) reliability and security, especially for enterprise and multi-instance setups.

- OIDC and SAML sync in multi-main setups [version: 1.113.0]: In multi-main deployments, updates to SSO settings are now synchronized across all instances, ensuring consistent login behavior everywhere.
- Enhanced OIDC integration [version 1.111.0]: n8n now supports OIDC providers that enforce state and nonce parameters. These are validated during login, providing smoother and more secure Single Sign-On.

### Filter insights by project

We've added project filtering to insights, enabling more granular reporting and visibility into individual project performance.

[Image: Insights]  
Filter insights

### Contributors

[ongdisheng](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.112.3

[View the commits](#) for this version.

**Release date:** 2025-09-19

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.111.1

[View the commits](#) for this version.

**Release date:** 2025-09-19

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.110.2

[View the commits](#) for this version.

**Release date:** 2025-09-19

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.112.2

[View the commits](#) for this version.

**Release date:** 2025-09-18

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.112.1

[View the commits](#) for this version.

**Release date:** 2025-09-17

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.112.0

[View the commits](#) for this version.

**Release date:** 2025-09-15

This release contains API improvements, core updates, editor improvements, node updates, and bug fixes.

### Additional API Endpoints versions

We've made several updates to the Executions API:

- Execution details: GET /executions now includes *status* and *workflow\_name* in the response.
- Retry execution endpoint: Added new public API endpoints to retry failed executions.
- Additional filters: You can now filter executions by running or canceled status.

## Enhancements to workflow diff

We added a several updates on workflows diffs as well:

- Better view in Code nodes and Stickies: Workflow diffs now highlight changes per line instead of per block, making edits easier to review and understand.
- Enable/Disable sync: You can now enable or disable sync in the viewport, letting you compare a workflow change in one view without affecting the other.

[Image: Workflow diff]  
Workflow diff

## Contributors

[GuraaseesSingh](#)  
[jabbson](#)  
[ongdisheng](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.111.0

View the [commits](#) for this version.

**Release date:** 2025-09-08

This release contains core updates, API improvements, node updates, and bug fixes.

## Contributors

[abellion](#)  
[cesars-gh](#)  
[durran](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.110.1

View the [commits](#) for this version.

**Release date:** 2025-09-03

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.109.2

View the [commits](#) for this version.

**Release date:** 2025-09-03

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.110.0

View the [commits](#) for this version.

**Release date:** 2025-09-01

This release contains core updates, editor improvements, node updates, performance improvements, and bug fixes.

### Contributors

[heyxmirko](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.109.1

View the [commits](#) for this version.

**Release date:** 2025-08-27

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.108.2

View the [commits](#) for this version.

**Release date:** 2025-08-27

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.109.0

View the [commits](#) for this version.

**Release date:** 2025-08-25

This release contains core updates, editor improvements, node updates, performance improvements, and bug fixes.

### Contributors

[naXa777](#)

[prettycode2022](#)

[oppai](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.107.4

View the [commits](#) for this version.

**Release date:** 2025-08-20

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.108.1

View the [commits](#) for this version.

**Release date:** 2025-08-20

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.107.3

View the [commits](#) for this version.

**Release date:** 2025-08-18

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.108.0

View the [commits](#) for this version.

**Release date:** 2025-08-18

This release contains a new CLI tool, editor improvements, node updates, performance improvements, and bug fixes. **Workflow Diff**

For teams working across different environments, deployments often involve multiple people making changes at different times. Without a clear view of those changes, it's easy to miss something important.

[Workflow Diff](#) gives you an easy and visual way to review workflow changes before you deploy them between environments.

With it, you can:

- Quickly see what's been added, changed, or deleted, with clear colour highlights.
- Easily see important settings changes on a workflow.
- Check changes inside each node, and spot connector updates, with a side-by-side view of its settings.
- Get a quick count of all changes to understand the size of a deployment.

[Image: Workflow diffs]

[Image: ]  
Workflow Diff

Workflow Diff eases the review and approval of changes before deployment, enabling teams to collaborate on workflows without breaking existing automations or disrupting production. It's one step further in integrating DevOps best practices in n8n.

Now available for Enterprise customers using Environments.

## Contributors

[ManuLasker](#)  
[EternalDeiwo](#)  
[jreyesr](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.107.2

View the [commits](#) for this version.

**Release date:** 2025-08-15

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.107.1

View the [commits](#) for this version.

**Release date:** 2025-08-14

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.106.3

View the [commits](#) for this version.

**Release date:** 2025-08-11

This release contains a backported update.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.107.0

View the [commits](#) for this version.

**Release date:** 2025-08-11

This release contains bug fixes.

## Contributors

[Amsal1](#)  
[andrewzolotukhin](#)  
[DMA902](#)  
[fkowal](#)  
[Gulianrdgd](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.106.2

View the [commits](#) for this version.

**Release date:** 2025-08-08

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.106.1

View the [commits](#) for this version.

**Release date:** 2025-08-07

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.105.4

View the [commits](#) for this version.

**Release date:** 2025-08-07

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.105.3

View the [commits](#) for this version.

**Release date:** 2025-08-05

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.106.0

View the [commits](#) for this version.

**Release date:** 2025-08-04

This release contains performance improvements, core updates, editor improvements, node updates, a new node, and bug fixes.

## No more limit of active workflows and new self-hosted Business Plan

We have rolled out a new pricing model to make it easier for builders of all sizes to adopt and scale automation with n8n.

### ***What's new***

#### **No more limit of active workflows.**

All n8n plans, from Starter to Enterprise, now include unlimited users, workflows, and steps. Our pricing is based on the volume of executions. Meaning you can build and test as many workflows as you want, including complex, data-heavy, or long-running automations, without worrying about quotas.

#### **New self-hosted Business Plan for growing teams**

Designed for SMBs and mid-sized companies, the Business Plan includes features such as:

- 6 shared projects
- SSO, SAML and LDAP
- Different environments
- Global variables
- Version control using Git
- 30 days of Insights

Please note that this plan only includes support from our community forum. For dedicated support we recommend upgrading to our Enterprise plan.

#### **Enterprise pricing now scales with executions**

Enterprise plans no longer use workflow-based pricing, and is now also based on the volume of executions.

### ***What you need to do***

To ensure these changes apply to your account, update your n8n instance to the latest version.

[Read the blog](#) for full details.

## Contributors

[baruchiro](#)  
[killthekitten](#)  
[baileympearson](#)  
[Yingrjimsch](#)  
[joshualipman123](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.105.2

View the [commits](#) for this version.

**Release date:** 2025-08-01

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.105.1

View the [commits](#) for this version.

**Release date:** 2025-08-01

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.104.2

View the [commits](#) for this version.

**Release date:** 2025-07-31

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.105.0

View the [commits](#) for this version.

**Release date:** 2025-07-28

This release contains core updates, editor improvements, node updates, and bug fixes.

### Respond to Chat node

With the [\\*\\*Respond to Chat node\\*\\*](#), you can now access Human-in-the-Loop functionality natively in n8n Chat.

Enable conversational experiences where you can ask for clarification, request approval before taking further action, and get back intermediate results — all within a single workflow execution.

This unlocks multi-turn interactions that feel more natural and reduce the number of executions required. It is ideal for building interactive AI use cases like conversational forms, branched workflows based on user replies, and step-by-step approvals.

#### \* How to:

- Add a **Chat Trigger** node and select **Using Respond Nodes** for the **Response mode**
- Place the **Respond to Chat** node anywhere in your workflow to send a message into the Chat and optionally wait for the user to input a response before continuing execution of the workflow steps.

## Contributors

[dana-gill](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.104.1

View the [commits](#) for this version.

**Release date:** 2025-07-23

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.103.2

View the [commits](#) for this version.

**Release date:** 2025-07-22

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.104.0

View the [commits](#) for this version.

**Release date:** 2025-07-21

=====

This release contains core updates, editor improvements, a new node, node updates, and bug fixes.

## Contributors

[nunulk](#)

[iaptsiauri](#)

[KGuillaume-chaps](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.101.3

View the [commits](#) for this version.

**Release date:** 2025-07-18

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.102.4

View the [commits](#) for this version.

**Release date:** 2025-07-17

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.103.1

View the [commits](#) for this version.

**Release date:** 2025-07-17

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.102.3

View the [commits](#) for this version.

**Release date:** 2025-07-14

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.103.0

View the [commits](#) for this version.

**Release date:** 2025-07-14

This release contains core updates, editor improvements, new nodes, node improvements, and bug fixes.

### Chat streaming

No more waiting for full responses to load when using the n8n chat interface. **Streaming** now delivers AI-generated text replies word by word so users can read messages as they're being generated. It feels faster, smoother, and more like what people expect from chat experiences.

Streaming is available in public chat views (hosted or embedded) and can be used in custom apps via webhook.

#### ⌘ How-to

Configure streaming in the Node Details View of these nodes:

- Chat Trigger node: Options>Add Field>Response Mode>Streaming
- Webhook node: Respond>Streaming
- AI Agent node: Add option> Enable streaming

## Improved instance user list with more visibility

The instance user list has been updated with a new table layout and additional details to help admins manage access more easily.

You can now:

- See total users and filter by name or email
- View which projects each user has access to
- Whether a user has enabled 2FA and sort based on that
- See the last active date for each user

This makes it easier to audit user activity, identify inactive accounts, and understand how access is distributed across your instance.

[Image: Improved instance user list with more visibility screenshot]

## Webhook HTML responses

Starting with this release, if your workflow sends an HTML response to a webhook, n8n automatically wraps the content in an `

### Evaluation node and tab

The **Evaluation node** includes several operations that, when used together, enable end-to-end AI evaluation.

[Image: Evaluation node]  
Evaluation node

Use this node to:

- Run your AI logic against a wide range of test cases in the same execution
- Capture the outputs of those test cases
- Score the results using your own metrics or LLM-as-judge logic
- Isolate a testing path to only include the nodes and logic you want to evaluate

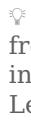
The **Evaluations tab** enables you to review test results in the n8n UI, perfect for comparing runs, spotting regressions, and viewing performance over time.

### ⌘ How evaluations work

The evaluation path runs alongside your normal execution logic and only activates when you want—making it ideal for testing and iteration.

Get started by selecting an AI workflow you want to evaluate that includes one or more LLM or Agent nodes.

1. Add an **Evaluation** node with the **On new Evaluation event** operation. This node will act as an additional trigger you'll run only when testing. Configure it to read your dataset from Google Sheets, with each row representing a test input.

 Better datasets mean better evaluations. Craft your dataset from a variety of test cases, including edge cases and typical inputs, to get meaningful feedback on how your AI performs. Learn more and access sample datasets [here](#).

2. Add a second **Evaluation** node using the **Set Outputs** operation after the part of the workflow you're testing—typically after an LLM or Agent node. This captures the response and writes it back to your dataset in Google Sheets.
3. To evaluate output quality, add a third **Evaluation** node with the **Set Metrics** operation at a point after you've generated the outputs. You can develop workflow logic, custom calculations, or add an LLM-as-Judge to score the outputs. Map these metrics to your dataset in the node's parameters.

 Well-defined metrics = smarter decisions. Scoring your outputs based on similarity, correctness, or categorization can help you track whether changes are actually improving performance. Learn more and get links to example templates [here](#).

[Image: Evaluation workflow]  
Evaluation workflow

When the Evaluation trigger node is executed, it runs each input in our dataset through your AI logic. This continues until all test cases are processed, a limit is reached, or you manually stop the execution. Once your evaluation path is set up, you can update your prompt, model, or workflow logic—and re-run the Evaluation trigger node to compare results. If you've added metrics, they'll appear in the Evaluations tab.

In some instances, you may want to isolate your testing path to make iteration faster or to avoid executing downstream logic. In this case, you can add an Evaluation node with the `Check If Evaluating` operation to ensure only the expected nodes run when performing evaluations.

### Things to keep in mind

Evaluations for AI Workflows are designed to fit into your development flow, with more enhancements on the way. For now, here are a few things to note:

- Test datasets are currently managed through Google Sheets. You'll need a Google Sheets credential to run evaluations.
- Each workflow supports one evaluation at a time. If you'd like to

test multiple segments, consider splitting them into sub-workflows for more flexibility.

- Community Edition supports one single evaluation. Pro and Enterprise plans allow unlimited evaluations.
- AI Evaluations are not enabled for instances in scaling mode at this time.

You can find details, tips, and common troubleshooting info [here](#).

☞ Learn more about the AI evaluation strategies and practical implementation techniques. [Watch now](#).

## Contributors

[Phiph](#)  
[cesars-gh](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.94.0

View the [commits](#) for this version.

**Release date:** 2025-05-19

This release contains editor improvements, an API update, node updates, new nodes, and bug fixes.

## Verified community nodes on Cloud

We've expanded the n8n ecosystem and unlocked a new level of flexibility for all users including those on n8n Cloud! Now you can access a select set of community nodes and partner integrations without leaving the canvas. This means you install and automate with a wider range of integrations without leaving your workspace. The power of the community is now built-in.

This update focuses on three major improvements:

- **Cloud availability:** Community nodes are no longer just for self-hosted users. A select set of nodes is now available on n8n Cloud.
- **Built-in discovery:** You can find and explore these nodes right from the Nodes panel without leaving the editor or searching on npm.
- **Trust and verification:** Nodes that appear in the editor have been manually vetted for quality and security. These verified nodes are marked with a checkmark.

We're starting with a selection of around 25 nodes, including some of the most-used community-built packages and partner-supported integrations. For this phase, we focused on nodes that don't include external package dependencies - helping streamline the review process and ensure a smooth rollout.

This is just the start. We plan to expand the library gradually, bringing even more verified nodes into the editor along with the powerful and creative use cases they unlock. In time, our criteria will evolve,

opening the door to a wider range of contributions while keeping quality and security in focus.

Learn more about this update and find out which nodes are already installable from the editor in our [blog](#) post.

### **☛ Use a verified node**

Make sure you're on **n8n version 1.94.0** or later and the instance Owner has enabled verified community nodes. On Cloud, this can be done from the Admin Panel. For self-hosted instances, please refer to [documentation](#). In both cases, verified nodes are enabled by default.

- Open the **Nodes panel** from the editor
- Search for the Node. Verified nodes are indicated by a shield 
- Select the node and click **Install**

Once an Owner installs a node, everyone on the instance can start using it—just drag, drop, and connect like any other node in your workflow.

### **\* Build a node and get it verified**

Want your node to be verified and discoverable from the editor? Here's how to get involved:

1. Review the [community node verification guidelines](#).
2. If you're building something new, follow the recommendations for [creating nodes](#).
3. Check your design against the [UX guidelines](#).
4. [Submit your node](#) to npm.
5. Request verification by filling out [this form](#).

### **Already built a node? Raise your hand!**

If you've already published a community node and want it considered for verification, make sure it meets the requirements noted above, then let us know by submitting the [interest form](#). We're actively curating the next batch and would love to include your work.

### **Extended logs view**

When workflows get complex, debugging can get... clicky. That's where an extended **Logs View** comes in. Now you can get a clearer path to trace executions, troubleshoot issues, and understand the behavior of a complete workflow — without bouncing between node detail views.

This update brings a unified, always-accessible panel to the bottom of the canvas, showing you each step of the execution as it happens. Whether you're working with loops, sub-workflows, or AI agents,

you'll see a structured view of everything that ran, in the order it ran—with input, output, and status info right where you need it.

You can jump into node details when you want to dig deeper, or follow a single item through every step it touched. Real-time highlighting shows you which nodes are currently running or have failed, and you'll see total execution time for any workflow—plus token usage for AI workflows to help monitor performance. And if you're debugging across multiple screens? Just pop the logs out and drag them wherever you'd like.

### ⌚What it does

- Adds a **Logs view** to the bottom of the canvas that can be opened or collapsed. (Chat also appears here if your workflow uses it).
- Displays a **hierarchical list of nodes** in the order they were executed—including expanded views of sub-workflows.
- Allows you to **click a node in hierarchy** to preview inputs and outputs directly, or jump into the full Node Details view with a link.
- Provides ability to **toggle** input and output data on and off.
- Highlights each node **live as it runs**, showing when it starts, completes, or fails.
- Includes **execution history** view to explore past execution data in a similar way.
- Shows **roll-up stats** like total execution time and total AI tokens used (for AI-enabled workflows).
- Includes a “**pop out**” button to open the logs as a floating window—perfect for dragging to another screen while debugging.

### 🛠How to

To access the expanded logs view, click on the Logs bar at the bottom of the canvas. The view is also opens up when you open the chat window on the bottom of the page.

## Contributors

[Stamsy](#)  
[feelgood-interface](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.93.0

View the [commits](#) for this version.

**Release date:** 2025-05-12

This release contains core updates, editor improvements, new nodes, node updates, and bug fixes.

### Faster ways to open sub-workflows

We've added several new ways to navigate your multi-workflow automations faster.

From any workflow with a sub-workflow node:

✍ Right-click on a sub-workflow node and select `Open sub-workflow` from the context menu

## 💻 Keyboard shortcuts

- **Windows:** CTRL + SHIFT + 0 or CTRL + Double Click
- **Mac:** CMD + SHIFT + 0 or CMD + Double Click

These options will bring your sub-workflow up in a new tab.

## Archive workflows

If you've ever accidentally removed a workflow, you'll appreciate the new archiving feature. Instead of permanently deleting workflows with the Remove action, workflows are now archived by default. This allows you to recover them if needed.

### How to:

- **Archive a workflow** - Select **Archive** from the Editor UI menu. It has replaced the **Remove** action.
- **Find archived workflows** - Archived workflows are hidden by default. To find your archived workflows, select the option for **Show archived workflows** in the workflow filter menu.
- **Permanently delete a workflow** - Once a workflow is archived, you can **Delete** it from the options menu.
- **Recover a workflow** - Select **Unarchive** from the options menu.

### Keep in mind:

- Workflows archival requires the same permissions as required previously for removal.
- You cannot select archived workflows as sub-workflows to execute
- Active workflows are deactivated when they are archived
- Archived workflows can not be edited

## Contributors

[LeaDevelop](#)  
[ayhandoslu](#)  
[valentina98](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.92.2

View the [commits](#) for this version.

**Release date:** 2025-05-08

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.91.3

View the [commits](#) for this version.

**Release date:** 2025-05-08

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.92.1

View the [commits](#) for this version.

**Release date:** 2025-05-06

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.92.0

View the [commits](#) for this version.

**Release date:** 2025-05-05

This release contains core updates, editor improvements, node updates, and bug fixes.

### Partial Execution for AI Tools

We've made it easier to build and iterate on AI agents in n8n. You can now run and test specific tools without having to execute the entire agent workflow.

Partial execution is especially useful when refining or troubleshooting parts of your agent logic. It allows you to test changes incrementally, without triggering full agent runs, reducing unnecessary AI calls, token usage, and downstream activity. This makes iteration faster, more cost-efficient, and more precise when working with complex or multi-step AI workflows.

Partial execution for AI tools is available now for all tools - making it even easier to build, test, and fine-tune AI agents in n8n.

### How to:

To use this feature you can either:

- Click the **Play** button on the tool you want to execute directly from the canvas view.
- Open the tool's **Node Details View** and select "**Execute Step**" to run it from there.

If you have previously run the workflow, the input and output will be prefilled with data from the last execution. A pop-up form will open where you can manually fill in the parameters before executing your test.

### Extended logs view

When workflows get complex, debugging can get... clicky. That's where an extended **Logs View** comes in. Now you can get a clearer path to trace executions, troubleshoot issues, and understand the behavior of a complete workflow — without bouncing between node detail views.

This update brings a unified, always-accessible panel to the bottom of the canvas, showing you each step of the execution as it happens. Whether you're working with loops, sub-workflows, or AI agents, you'll see a structured view of everything that ran, in the order it ran—with input, output, and status info right where you need it.

You can jump into node details when you want to dig deeper, or follow a single item through every step it touched. Real-time highlighting shows you which nodes are currently running or have failed, and you'll see total execution time for any workflow—plus token usage for AI workflows to help monitor performance. And if you're debugging across multiple screens? Just pop the logs out and drag them wherever you'd like.

### ⌚What it does

- Adds a **Logs view** to the bottom of the canvas that can be opened or collapsed. (Chat also appears here if your workflow uses it).
- Displays a **hierarchical list of nodes** in the order they were executed—including expanded views of sub-workflows.
- Allows you to **click a node in hierarchy** to preview inputs and outputs directly, or jump into the full Node Details view with a link.
- Provides ability to **toggle** input and output data on and off.
- Highlights each node **live as it runs**, showing when it starts, completes, or fails.
- Includes **execution history** view to explore past execution data in a similar way.
- Shows **roll-up stats** like total execution time and total AI tokens used (for AI-enabled workflows).
- Includes a “**pop out**” button to open the logs as a floating window —perfect for dragging to another screen while debugging.

### ⌘How to

To access the expanded logs view, click on the Logs bar at the bottom of the canvas. The view is also opens up when you open the chat window on the bottom of the page.

## Insights enhancements for Enterprise

Two weeks after the launch of [Insights](#), we're releasing some enhancements designed for enterprise users.

- **Expanded time ranges.** You can now filter insights over a variety of time periods, from the last 24 hours up to 1 year. Pro users are limited to 7 day and 14 day views.
- **Hourly granularity.** Drill down into the last 24 hours of production executions with hourly granularity, making it easier to analyze workflows and quickly identify issues.

These updates provide deeper visibility into workflow history, helping you uncover trends over longer periods and detect problems sooner with more precise reporting.

[Image: Filter insights]  
Filter insights

## Contributors

[Stamsy](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.91.2

View the [commits](#) for this version.

**Release date:** 2025-05-05

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.90.3

View the [commits](#) for this version.

**Release date:** 2025-05-05

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.91.1

View the [commits](#) for this version.

**Release date:** 2025-05-01

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.91.0

View the [commits](#) for this version.

**Release date:** 2025-04-28

This release contains core updates, editor improvements, node updates, and bug fixes.

### Breadcrumb view from the canvas

We've added **breadcrumb navigation directly on the canvas**, so you can quickly navigate to any of a workflow's parent folders right from the canvas.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.90.2

View the [commits](#) for this version.

**Release date:** 2025-04-25

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.90.1

View the [commits](#) for this version.

**Release date:** 2025-04-22

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.90.0

View the [commits](#) for this version.

**Release date:** 2025-04-22

This release contains core updates, editor updates, node updates, performance improvements, and bug fixes.

### Extended HTTP Request tool functionality

We've brought the full power of the HTTP Request node to the HTTP Request tool in AI workflows. That means your AI Agents now have access to all the advanced configuration options—like Pagination, Batching, Timeout, Redirects, Proxy support, and even cURL import.

This update also includes support for the `$fromAI` function to dynamically generate the right parameters based on the context of your prompt — making API calls smarter, faster, and more flexible than ever.

#### How to:

- Open your AI Agent node in the canvas.
  - Click the '+' icon to add a new tool connection.
  - In the **Tools panel**, select **HTTP Request Tool**.
  - Configure it just like you would a regular **HTTP Request node** — including advanced options
- ☞ Learn more about configuring the [HTTP Request tool](#).

### Scoped API keys

Users on the Enterprise plan can now create API keys with specific scopes to control exactly what each key can access.

[Image: Scoped API keys]  
Scoped API keys

Previously, API keys had full read/write access across all endpoints. While sometimes necessary, this level of access can be excessive and too powerful for most use cases. Scoped API keys allow you to limit

access to only the resources and actions a service or user actually needs.

## What's new

When creating a new API key, you can now:

- Select whether the key has read, write, or both types of access.
- Specify which resources the key can interact with.

Supported scopes include:

- Variables — list, create, delete
- Security audit — generate reports
- Projects — list, create, update, delete
- Executions — list, read, delete
- Credentials — list, create, update, delete, move
- Workflows — list, create, update, delete, move, add/remove tags

Scoped API keys give you more control and security. You can limit access to only what's needed, making it safer to work with third parties and easier to manage internal API usage.

## Drag and Drop in Folders

Folders just got friendlier. With this release, you can now **drag and drop workflows and folders** — making it even easier to keep things tidy.

Need to reorganize? Just select a workflow or folder and drag it into another folder or breadcrumb location. It's a small change that makes a big difference when managing a growing collection of workflows.

↳ Folders are available to all registered users—jump in and get your workspace in order!

## Contributors

Zordrak

For full release details, refer to Releases on GitHub.

## n8n@1.89.2

View the commits for this version.

**Release date:** 2025-04-16

This release contains a bug fix.

For full release details, refer to Releases on GitHub.

## n8n@1.89.1

View the [commits](#) for this version.

**Release date:** 2025-04-15

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.89.0

View the [commits](#) for this version.

**Release date:** 2025-04-14

This release contains API updates, core updates, editor updates, a new node, node updates, and bug fixes.

## Insights

We're rolling out [Insights](#), a new dashboard to monitor how your workflows are performing over time. It's designed to give admins (and owners) better visibility of their most important workflow metrics and help troubleshoot potential issues and improvements.

In this first release, we're introducing a summary banner, the insights dashboard, and time saved per execution.

### 1. Summary banner

A new banner on the overview page that gives instance admins and owners a birds eye view of key metrics over the last 7 days.

[Image: Summary banner]  
Insights summary banner

Available metrics:

- Total production executions
- Total failed executions
- Failure rate
- Average runtime of all workflows
- Estimated time saved

This overview is designed to help you stay on top of workflow activity at a glance. It is available for all plans and editions.

### 2. Insights dashboard

On Pro and Enterprise plans, a new dashboard offers a deeper view into workflow performance and activity.

[Image: Insights dashboard]  
Insights dashboard

The dashboard includes:

- Total production executions over time, including a comparison of successful and failed executions

- Per-workflow breakdowns of key metrics
- Comparisons with previous periods to help spot changes in usage or behavior
- Runtime average and failure rate over time

### **3. Time saved per execution**

Within workflow settings, you can now assign a “time saved per execution” value to any workflow. This makes it possible to track the impact of your workflows and make it easier to share this visually with other teams and stakeholders.

This is just the beginning for Insights: the next phase will introduce more advanced filtering and comparisons, custom date ranges, and additional monitoring capabilities.

### **Node updates**

- We added a credential check for the Salesforce node
- We added SearXNG as a tool for AI agents

You can now search within subfolders, making it easier to find workflows across all folder levels. Just type in the search bar and go.

For full release details, refer to [Releases](#) on GitHub.

## **n8n@1.88.0**

View the [commits](#) for this version.

**Release date:** 2025-04-10

This release contains new features, new nodes, performance improvements, and bug fixes.

### **Model Context Protocol (MCP) nodes**

MCP aims to standardise how LLMs like Claude, ChatGPT, or Cursor can interact with tools or integrate data for their agents. Many providers - both established or new - are adopting MCP as a standard way to build agentic systems. It is an easy way to either expose your own app as a server, making capabilities available to a model as tools, or as a client that can call on tools outside of your own system.

While it's still early in the development process, we want to give you access to our new MCP nodes. This will help us understand your requirements better and will also let us converge on a great general solution quicker.

We are adding two new nodes:

- a MCP [Server Trigger](#) for any workflow
- a MCP [Client Tool](#) for the AI Agent

The MCP Server Trigger turns n8n into an MCP server, providing n8n tools to models running outside of n8n. You can run multiple MCP servers from your n8n instance. The MCP Client Tool connects LLMs - and other intelligent agents - to any MCP-enabled service through a single interface.

Max from our DevRel team created an official walkthrough for you to get started:

[Image: Studio]

[Studio Update #04](#)

## MCP Server Trigger

The MCP Server Trigger turns n8n into an MCP server, providing n8n tools to models running outside of n8n. The node acts as an entry point into n8n for MCP clients. It operates by exposing a URL that MCP clients can interact with to access n8n tools. This means your n8n workflows and integrations are now available to models run elsewhere. Pretty neat.

[Image: MCP Server Trigger]  
MCP Server Trigger

[Explore the MCP Server Trigger docs](#)

## MCP Client Tool

The MCP Client Tool node is a MCP client, allowing you to use the tools exposed by an external MCP server. You can connect the MCP Client Tool node to your models to call external tools with n8n agents. In this regard it is similar to using a n8n tool with your AI agent. One advantage is that the MCP Client Tool can access multiple tools on the MCP server at once, keeping your canvas cleaner and easier to understand.

[Image: MCP Client Tool]  
MCP Client Tools

[Explore the MCP Client Tool docs](#)

## Node updates

- Added a node for Azure Cosmos DB
- Added a node for Milvus Vector Store
- Updated the Email Trigger (IMAP) node

## Contributors

[adina-hub](#)  
[umanamente](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.87.2

View the [commits](#) for this version.

**Release date:** 2025-04-09

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.86.1

View the [commits](#) for this version.

**Release date:** 2025-04-09

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.87.1

View the [commits](#) for this version.

**Release date:** 2025-04-08

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.87.0

View the [commits](#) for this version.

**Release date:** 2025-04-07

This release contains new nodes, node updates, API updates, core updates, editor updates, and bug fixes.

### Contributors

[cesars-gh](#)

[Stamsy](#)

[Pash10q](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.86.0

View the [commits](#) for this version.

**Release date:** 2025-03-31

This release contains API updates, core updates, editor improvements, node updates, and bug fixes.

### Contributors

[Aijeyomah](#)

[ownerer](#)

[ulevitsky](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.85.4

[View the commits](#) for this version.

**Release date:** 2025-03-27

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.84.3

[View the commits](#) for this version.

**Release date:** 2025-03-27

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.84.2

[View the commits](#) for this version.

**Release date:** 2025-03-26

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.85.3

[View the commits](#) for this version.

**Release date:** 2025-03-26

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.85.2

[View the commits](#) for this version.

**Release date:** 2025-03-25

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.85.1

[View the commits](#) for this version.

**Release date:** 2025-03-25

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.85.0

View the [commits](#) for this version.

**Release date:** 2025-03-24

This release contains a new node, a new credential, core updates, editor updates, node updates, and bug fixes.

## Folders

What can we say about folders? Well, they're super handy for categorizing just about everything and they're finally available for your n8n workflows. Tidy up your workspace with unlimited folders and nested folders. Search for workflows within folders. It's one of the ways we're making it easier to organize your n8n instances more effectively.

### How to use it:

Create and manage folders within your personal space or within projects. You can also create workflows from within a folder. You may need to restart your instance in order to activate folders.

[Image: Folders]  
It's a folder alright

Folders are available for all [registered](#) users so get started with decluttering your workspace now and look for more features (like drag and drop) to organize your instances soon.

## Enhancements to Form Trigger Node

Recent updates to the Form Trigger node have made it a more powerful tool for building business solutions. These enhancements provide more flexibility and customization, enabling teams to create visually engaging and highly functional workflows with forms.

- **HTML customization:** Add custom HTML to forms, including embedded images and videos, for richer user experiences.
- **Custom CSS support:** Apply custom styles to user-facing components to align forms with your brand's look and feel. Adjust fonts, colors, and spacing for a seamless visual identity.
- **Form previews:** Your form's description and title will pull into previews of your form when sharing on social media or messaging apps, providing a more polished look.
- **Hidden fields:** Use query parameters to add hidden fields, allowing you to pass data—such as a referral source—withou exposing it to the user.
- **New responses options:** Respond to user submissions in multiple ways including text, HTML, or a downloadable file (binary format). This enables forms to display rich webpages or deliver digital assets such as dynamically generated invoices or personalized certificates.

[Image: Form with custom CSS applied]  
Form with custom CSS applied

These improvements elevate the Form Trigger node beyond a simple workflow trigger, transforming it into a powerful tool for addressing use cases from data collection and order processing to custom content creation.

## Contributors

[Fank](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.84.1

View the [commits](#) for this version.

**Release date:** 2025-03-18

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.84.0

View the [commits](#) for this version.

**Release date:** 2025-03-17

This release contains a new node, node updates, editor updates, and bug fixes.

## Contributors

[Pash10g](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.83.2

View the [commits](#) for this version.

**Release date:** 2025-03-14

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.82.4

View the [commits](#) for this version.

**Release date:** 2025-03-14

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.82.3

View the [commits](#) for this version.

**Release date:** 2025-03-13

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.83.1

View the [commits](#) for this version.

**Release date:** 2025-03-12

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.83.0

View the [commits](#) for this version.

**Release date:** 2025-03-12

This release contains bug fixes and an editor update.

### Schema Preview

Schema Preview lets you view and work with a node's expected output without executing it or adding credentials, keeping you in flow while building.

- **See expected node outputs instantly.** View schemas for over 100+ nodes to help you design workflows efficiently without extra steps.
- **Define workflow logic first, take care of credentials later.** Build your end-to-end workflow without getting sidetracked by credential setup.
- **Avoid unwanted executions when building.** Prevent unnecessary API calls, unwanted data changes, or potential third-party service costs by viewing outputs without executing nodes.

#### How to use it:

- Add a node with Schema Preview support to your workflow.
- Open the next node in the sequence - Schema Preview data appears in the Node Editor where you would typically find it in the Schema View.
- Use Schema Preview fields just like other schema data - drag and drop them into parameters and settings as needed.

Don't forget to add the required credentials before putting your workflow into production.

### Contributors

[pemontto](#)  
[Haru922](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.82.2

View the [commits](#) for this version.

**Release date:** 2025-03-12

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.82.1

View the [commits](#) for this version.

**Release date:** 2025-03-04

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.82.0

View the [commits](#) for this version.

**Release date:** 2025-03-03

This release contains core updates, editor updates, new nodes, node updates, new credentials, credential updates, and bug fixes.

### Tidy up

Tidy up instantly aligns nodes, centers stickies, untangles connections, and brings structure to your workflows. Whether you're preparing to share a workflow or just want to improve readability, this feature saves you time and makes your logic easier to follow. Clean, well-organized workflows aren't just nicer to look at—they're also quicker to understand.

#### How to:

Open the workflow you want to tidy, then choose one of these options:

- Click the **Tidy up** button in the bottom-left corner of the canvas (it looks like a broom 
- Press **Shift + Alt + T** on your keyboard
- Right-click anywhere on the canvas and select **Tidy up workflow**

Want to tidy up just part of your workflow? Select the specific nodes you want to clean up first - Tidy up will only adjust those, along with any stickies behind them.

## Multiple API keys

n8n now supports multiple API keys, allowing users to generate and manage separate keys for different workflows or integrations. This improves security by enabling easier key rotation and isolation of credentials. Future updates will introduce more granular controls.

[Image: Multiple API keys]  
Multiple API keys

## Contributors

[RostamMahabadi](#)  
[Lanhild](#)  
[matthiez](#)  
[feelgood-interface](#)  
[adina-hub](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.81.4

View the [commits](#) for this version.

**Release date:** 2025-03-03

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.81.3

View the [commits](#) for this version.

**Release date:** 2025-03-03

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.81.2

View the [commits](#) for this version.

**Release date:** 2025-02-28

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.80.5

View the [commits](#) for this version.

**Release date:** 2025-02-28

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.80.4

View the [commits](#) for this version.

**Release date:** 2025-02-27

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.81.1

View the [commits](#) for this version.

**Release date:** 2025-02-27

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.81.0

View the [commits](#) for this version.

**Release date:** 2025-02-24

This release contains bug fixes, a core update, editor improvements, and a node update.

### Improved partial executions

The new execution engine for partial executions ensures that testing parts of a workflow in the builder closely mirrors production behaviour. This makes iterating with updated run-data faster and more reliable, particularly for complex workflows.

Before, user would test parts of a workflow in the builder that didn't consistently reflect production behaviour, leading to unexpected results during development.

This update aligns workflow execution in the builder with production behavior.

Here is an example for loops:

Before

After

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.80.3

View the [commits](#) for this version.

**Release date:** 2025-02-21

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.79.4

View the [commits](#) for this version.

**Release date:** 2025-02-21

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.80.2

View the [commits](#) for this version.

**Release date:** 2025-02-21

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.79.3

View the [commits](#) for this version.

**Release date:** 2025-02-21

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.80.1

View the [commits](#) for this version.

**Release date:** 2025-02-20

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.79.2

View the [commits](#) for this version.

**Release date:** 2025-02-20

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.80.0

View the [commits](#) for this version.

**Release date:** 2025-02-17

This release contains bug fixes and an editor improvement.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.75.3

View the [commits](#) for this version.

**Release date:** 2025-02-17

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.74.4

View the [commits](#) for this version.

**Release date:** 2025-02-17

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.79.1

View the [commits](#) for this version.

**Release date:** 2025-02-15

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.78.1

View the [commits](#) for this version.

**Release date:** 2025-02-15

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.77.4

View the [commits](#) for this version.

**Release date:** 2025-02-15

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.76.4

View the [commits](#) for this version.

**Release date:** 2025-02-15

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.79.0

View the [commits](#) for this version.

**Release date:** 2025-02-12

This release contains new features, node updates, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.77.3

View the [commits](#) for this version.

**Release date:** 2025-02-06

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.78.0

View the [commits](#) for this version.

**Release date:** 2025-02-05

This release contains new features, node updates, and bug fixes.

## Contributors

[mocanew](#)  
[Timtendo12](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.77.2

View the [commits](#) for this version.

**Release date:** 2025-02-04

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.76.3

View the [commits](#) for this version.

**Release date:** 2025-02-04

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.77.1

View the [commits](#) for this version.

**Release date:** 2025-02-03

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.76.2

View the [commits](#) for this version.

**Release date:** 2025-02-03

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.77.0

View the [commits](#) for this version.

**Release date:** 2025-01-29

This release contains new features, editor updates, new nodes, new credentials, node updates, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.76.1

View the [commits](#) for this version.

**Release date:** 2025-01-23

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.76.0

View the [commits](#) for this version.

**Release date:** 2025-01-22

This release contains new features, editor updates, new credentials, node improvements, and bug fixes.

## Contributors

[Stamsy](#)

[GKdeVries](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.75.2

[View the commits](#) for this version.

**Release date:** 2025-01-17

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.74.3

[View the commits](#) for this version.

**Release date:** 2025-01-17

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.75.1

[View the commits](#) for this version.

**Release date:** 2025-01-17

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.74.2

[View the commits](#) for this version.

**Release date:** 2025-01-17

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.75.0

[View the commits](#) for this version.

**Release date:** 2025-01-15

This release contains bug fixes and editor updates.

### Improved consistency across environments

We added new UX and automatic changes improvements resulting in a better consistency between your staging and production instances.

Previously, users faced issues like:

- Lack of visibility into required credential updates when pulling changes
- Incomplete synchronization, where changes — such as deletions — weren't always applied across environments
- Confusing commit process, making it unclear what was being pushed or pulled

We addressed these by:

- Clearly indicating required credential updates when pulling changes
- Ensuring deletions and other modifications sync correctly across environments
- Improving commit selection to provide better visibility into what's being pushed

[Image: Commit modal]  
Commit modal

[Image: Pull notification]  
Pull notification

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.74.1

View the [commits](#) for this version.

**Release date:** 2025-01-09

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.74.0

View the [commits](#) for this version.

**Release date:** 2025-01-08

This release contains new features, a new node, node updates, performance improvements and bug fixes.

### Overhauled Code node editing experience

We added a ton of new helpers to the Code node, making edits of your code much faster and more comfortable. You get:

- TypeScript autocomplete
- TypeScript linting
- TypeScript hover tips
- Search and replace
- New keyboard shortcuts based on the VSCode keymap
- Auto-formatting using prettier (Alt+Shift+F)
- Remember folded regions and history after refresh
- Multi cursor
- Type function in the Code node using JSDoc types
- Drag and drop for all Code node modes
- Indentation markers

We build this on a web worker architecture so you won't have to suffer from performance degradation while typing.

To get the full picture, check out our Studio update with Max and Elias, where they discuss and demo the new editing experience. 

[\[Image: Studio\]](#)

### [Studio Update #04](#)

## New node: Microsoft Entra ID

Microsoft Entra ID (formerly known as Microsoft Azure Active Directory or Azure AD) is used for cloud-based identity and access management. [The new node](#) supports a wide range of Microsoft Entra ID features, which includes creating, getting, updating, and deleting users and groups, as well as adding users to and removing them from groups.

## Node updates

- [AI Agent](#): Vector stores can now be directly used as tools for the agent
- [Code](#): Tons of new speed and convenience features, see above for details
- [Google Vertex Chat](#): Added option to specify the GCP region for the Google API credentials
- [HighLevel](#): Added support for calendar items

We also added a custom [projects](#) icon selector on top of the available emojis. Pretty!

## Contributors

[igatanasov](#)  
[Stamsy](#)  
[feelgood-interface](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.73.1

View the [commits](#) for this version.

**Release date:** 2024-12-19

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.73.0

View the [commits](#) for this version.

**Release date:** 2024-12-19

This release contains node updates, performance improvements, and bug fixes.

## Node updates

- [AI Agent](#): Updated descriptions for Chat Trigger options
- [Facebook Graph API](#): Updated for API v21.0
- [Gmail](#): Added two new options for the Send and wait operation, free

- text and custom form
- [Linear Trigger](#): Added support for admin scope
  - [MailerLite](#): Now supports the new API
  - [Slack](#): Added two new options for the Send and wait operation, free text and custom form

We also added credential support for [SolarWinds IPAM](#) and [SolarWinds Observability](#).

Last, but not least, we [improved the schema view performance in the node details view by 90%](#) and added drag and drop re-ordering to parameters. This comes in very handy in the [If](#) or [Edit Fields](#) nodes.

## Contributors

[CodeShakingSheep](#)  
[mickaelandrieu](#)  
[Stamsy](#)  
[pbdc0](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.72.1

View the [commits](#) for this version.

**Release date:** 2024-12-12

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.71.3

View the [commits](#) for this version.

**Release date:** 2024-12-12

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.72.0

View the [commits](#) for this version.

**Release date:** 2024-12-11

This release contains node updates, usability improvements, and bug fixes.

## Node updates

- [AI Transform](#): The maximum context length error now retries with reduced payload size
- [Redis](#): Added support for continue on fail

## Improved commit modal

We added filters and text search to the commit modal when working with [Environments](#). This will make committing easier as we provide more information and better visibility. Environments are available on the Enterprise plan.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.71.2

View the [commits](#) for this version.

**Release date:** 2024-12-10

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.70.4

View the [commits](#) for this version.

**Release date:** 2024-12-10

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.71.1

View the [commits](#) for this version.

**Release date:** 2024-12-06

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.70.3

View the [commits](#) for this version.

**Release date:** 2024-12-05

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.71.0

View the [commits](#) for this version.

**Release date:** 2024-12-04

This release contains node updates, performance improvements, and bug fixes.

## Task runners for the Code node in public beta

We're introducing a significant performance upgrade to the Code node with our new Task runner system. This enhancement moves JavaScript code execution to a separate process, improving your workflow execution speed while adding better isolation.

[Image: Task runners overview]

### Task runners overview

Our benchmarks show up to 6x improvement in workflow executions using Code nodes - from approximately 6 to 35 executions per second. All these improvements happen under the hood, keeping your Code node experience exactly the same.

The Task runner comes in two modes:

- Internal mode (default): Perfect for getting started, automatically managing task runners as child processes
- External mode: For advanced hosting scenarios requiring maximum isolation and security

Currently, this feature is opt-in and can be enabled using [environment variables](#). Once stable, it will become the default execution method for Code nodes.

To start using Task runners today, [check out the docs](#).

## Node updates

- [AI Transform node](#): We improved the prompt for code generation to transform data
- [Code node](#): We added a warning if `pairedItem` is absent or could not be auto mapped

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.70.2

View the [commits](#) for this version.

**Release date:** 2024-12-04

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.70.1

View the [commits](#) for this version.

**Release date:** 2024-11-29

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.70.0

View the [commits](#) for this version.

**Release date:** 2024-11-27

This release contains node updates, performance improvements and bug fixes.

## New canvas in beta

The new canvas is now the default setting for all users. It should bring significant performance improvements and adds a handy minimap. As it is still a beta version you can still revert to the previous version with the three dot menu.

We're looking forward to your feedback. Should you encounter a bug, you will find a handy button to create an issue at the bottom of the new canvas as well.

## Node updates

- We added credential support for [Zabbix](#) to the HTTP request node
- We added new OAuth2 credentials for [Microsoft SharePoint](#)
- The [Slack node](#) now uses markdown for the approval message when using the Send and Wait for Approval operation

## Contributors

[feelgood-interface](#)  
[adina-hub](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.68.1

View the [commits](#) for this version.

**Release date:** 2024-11-26

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.69.2

View the [commits](#) for this version.

**Release date:** 2024-11-26

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.69.1

View the [commits](#) for this version.

**Release date:** 2024-11-25

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.69.0

View the [commits](#) for this version.

**Release date:** 2024-11-20

This release contains a new feature, node improvements and bug fixes.

### Sub-workflow debugging

We made it much easier to debug sub-workflows by improving their accessibility from the parent workflow.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.68.0

View the [commits](#) for this version.

**Release date:** 2024-11-13

This release contains node updates, performance improvements and many bug fixes.

### New AI agent canvas chat

We revamped the chat experience for AI agents on the canvas. A neatly organized view instead of a modal hiding the nodes. You can now see the canvas, chat and logs at the same time when testing your workflow.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.67.1

View the [commits](#) for this version.

**Release date:** 2024-11-07

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.67.0

View the [commits](#) for this version.

**Release date:** 2024-11-06

This release contains node updates and bug fixes.

### Node updates

- [AI Transform](#): Improved usability

- [Anthropic Chat Model Node](#): Added Haiku 3.5 support
- [Convert to File](#): Added delimiter option for writing to CSV
- [Gmail Trigger](#): Added option to filter for draft messages
- [Intercom](#): Credential can now be used in the HTTP Request node
- [Rapid7 InsightVM](#): Added credential support

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.66.0

View the [commits](#) for this version.

**Release date:** 2024-10-31

This release contains performance improvements, a node update and bug fixes.

### Node update

- [Anthropic Chat Model](#): Added support for claude-3-5-sonnet-20241022

We made updates to how projects and workflow ownership are displayed making them easier to understand and navigate.

We further improved the performance logic of partial executions, leading to a smoother and more enjoyable building experience.

### New n8n canvas alpha

We have enabled the alpha version of our new canvas. The canvas is the ‘drawing board’ of the n8n editor, and we’re working on a full rewrite. Your feedback and testing will help us improve it. [Read all about it on our community forum](#).

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.65.2

View the [commits](#) for this version.

**Release date:** 2024-10-28

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.64.3

View the [commits](#) for this version.

**Release date:** 2024-10-25

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.65.1

View the [commits](#) for this version.

**Release date:** 2024-10-25

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.65.0

View the [commits](#) for this version.

**Release date:** 2024-10-24

This release contains a new features, new nodes, node enhancements, and bug fixes.

### New node: n8n Form

Use the [n8n Form node](#) to create user-facing forms with multiple pages. You can add other nodes with custom logic between to process user input. Start the workflow with a [n8n Form Trigger](#).

[Image: A multi-page form with branching]  
A multi-page form with branching

Additionally you can:

- Set default selections with query parameters
- Define the form with a JSON array of objects
- Show a completion screen and redirect to another URL

### Node updates

New nodes:

- [Google Business Profile](#) and [Google Business Profile Trigger](#): Use these to integrate Google Business Profile reviews and posts with your workflows

Enhanced nodes:

- [AI Agent](#): Removed the requirement to add at least one tool
- [GitHub](#): Added workflows as a resource operation
- [Structured Output Parser](#): Added more user-friendly error messages

For additional security, we improved how we handle multi-factor authentication, hardened config file permissions and introduced JWT for the public API.

For better performance, we improved how partial executions are handled in loops.

For full release details, refer to [Releases](#) on GitHub.

### Contributors

- [Idan Fishman](#)

## n8n@1.64.2

View the [commits](#) for this version.

**Release date:** 2024-10-24

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.64.1

View the [commits](#) for this version.

**Release date:** 2024-10-21

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.64.0

View the [commits](#) for this version.

**Release date:** 2024-10-16

This release contains a new node, node enhancements, performance improvements and bug fixes.

### Enhanced node: Remove Duplicates

The [Remove Duplicates node](#) got a major makeover with the addition of two new operations:

- Remove Items Processed in Previous Executions: Compare items in the current input to items from previous executions and remove duplicates
- Clear Deduplication History: Wipe the memory of items from previous executions.

This makes it easier to only process new items from any data source. For example, you can now more easily poll a Google sheet for new entries by id or remove duplicate orders from the same customer by comparing their order date. The great thing is, you can now do this within **and across** workflow runs.

### New node: Gong

The new node for [Gong](#) allows you to get users and calls to process them further in n8n. Very useful for sales related workflows.

For full release details, refer to [Releases](#) on GitHub.

### Contributors

- [Sören Uhrbach](#)

## n8n@1.63.4

[View the commits](#) for this version.

**Release date:** 2024-10-15

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.62.6

[View the commits](#) for this version.

**Release date:** 2024-10-15

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.63.3

[View the commits](#) for this version.

**Release date:** 2024-10-15

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.63.2

[View the commits](#) for this version.

**Release date:** 2024-10-11

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.62.5

[View the commits](#) for this version.

**Release date:** 2024-10-11

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.63.1

[View the commits](#) for this version.

**Release date:** 2024-10-11

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.62.4

View the [commits](#) for this version.

**Release date:** 2024-10-11

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.63.0

View the [commits](#) for this version.

**Release date:** 2024-10-09

This release contains new features, node enhancements and bug fixes.

### Node updates

- [OpenAI](#): Added the option to choose between the default memory connector to provide memory to the assistant or to specify a thread ID
- [Gmail](#) and [Slack](#): Added custom approval operations to have a human in the loop of a workflow

We have also optimized the [worker health checks](#) (see breaking change above).

Each credential now has a separate url you can link to. This makes sharing much easier.

For full release details, refer to [Releases](#) on GitHub.

### Contributors

[Pemontto](#)

## n8n@1.62.3

View the [commits](#) for this version.

**Release date:** 2024-10-08

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.62.2

View the [commits](#) for this version.

**Release date:** 2024-10-07

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.62.1

View the [commits](#) for this version.

**Release date:** 2024-10-02

This release contains new features, node enhancements and bug fixes.

### Additional nodes as tools

We have made additional nodes usable with the [Tools AI Agent node](#).

Additionally, we have added a `$fromAI()` placeholder function to use with tools, allowing you to dynamically pass information from the models to the connected tools. This function works similarly to placeholders used elsewhere in n8n.

Both of these new features enable you to build even more powerful AI agents by drawing directly from the apps your business uses. This makes integrating LLMs into your business processes even easier than before.

### Node updates

- [Google BigQuery](#): Added option to return numeric values as integers and not strings
- [HTTP Request](#): Added credential support for Sysdig
- [Invoice Ninja](#): Additional query params for getAll requests
- [Question and Answer Chain](#): Added the option to use a custom prompt

Drag and drop insertion on cursor position from schema view is now also enabled for code, SQL and Html fields in nodes.

Customers with an enterprise license can now rate, tag and highlight execution data in the executions view. To use highlighting, add an [Execution Data Node](#) (or Code node) to the workflow to set [custom executions data](#).

For full release details, refer to [Releases](#) on GitHub.

### Contributors

[Benjamin Roedell](#)

[CodeShakingSheep](#)

[manuelbcd](#)

[Miguel Prytoluk](#)

## n8n@1.61.0

View the [commits](#) for this version.

**Release date:** 2024-09-25

This release contains new features, node enhancements and bug fixes.

### Node updates

- [Brandfetch](#): Updated to use the new API

- [Slack](#): Made adding or removing the workflow link to a message easier

Big datasets now render faster thanks to virtual scrolling and execution annotations are harder to delete.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.59.4

View the [commits](#) for this version.

**Release date:** 2024-09-20

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.60.1

View the [commits](#) for this version.

**Release date:** 2024-09-20

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.60.0

View the [commits](#) for this version.

**Release date:** 2024-09-18

This release contains new features, node enhancements and bug fixes.

### Queue metrics for workers

You can now [expose and consume metrics from your workers](#). The worker instances have the same metrics available as the main instance(s) and can be configured with [environment variables](#).

You can now customize the maximum file size when uploading files within forms to webhooks. The [environment variable to set](#) for this is N8N\_FORMDATA\_FILE\_SIZE\_MAX. The default setting is 200MiB.

### Node updates

Enhanced nodes:

- [Invoice Ninja](#): Added actions for bank transactions
- [OpenAI](#): Added O1 models to the model select

For full release details, refer to [Releases](#) on GitHub.

### Contributors

[CodeShakingSheep](#)

## n8n@1.59.3

View the [commits](#) for this version.

**Release date:** 2024-09-18

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.59.2

View the [commits](#) for this version.

**Release date:** 2024-09-17

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.59.1

View the [commits](#) for this version.

**Release date:** 2024-09-16

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.58.2

View the [commits](#) for this version.

**Release date:** 2024-09-12

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.59.0

View the [commits](#) for this version.

**Release date:** 2024-09-11

This release contains bug fixes and feature enhancements.

For full release details, refer to [Releases](#) on GitHub.

## Contributors

[oscarpedrero](#)

## n8n@1.58.1

View the [commits](#) for this version.

**Release date:** 2024-09-06

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.58.0

View the [commits](#) for this version.

**Release date:** 2024-09-05

This release contains new features, bug fixes and feature enhancements.

### New node: PGVector Vector Store

This release adds the PGVector Vector Store node. Use this node to interact with the PGVector tables in your PostgreSQL database. You can insert, get, and retrieve documents from a vector table to provide them to a retriever connected to a chain.

### See active collaborators on workflows

We added collaborator avatars back to the workflow canvas. You will see other users who are active on the workflow, preventing you from overriding each other's work.

[Image: Collaboration avatars]

Collaboration avatars

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.57.0

View the [commits](#) for this version.

**Release date:** 2024-08-28

This release contains new features and bug fixes.

### Improved execution queue handling

We are [exposing new execution queue metrics](#) to give users more visibility of the queue length. This helps to inform decisions on horizontal scaling, based on queue status. We have also made querying executions faster.

### New credentials for the HTTP Request node

We added credential support for Datadog, Dynatrace, Elastic Security, Filescan, Iris, and Malcore to the HTTP Request node making it easier to use existing credentials.

We also made it easier to select workflows as tools when working with AI agents by implementing a new workflow selector parameter type.

For full release details, refer to [Releases](#) on GitHub.

## Contributors

[Bram Kn](#)

## n8n@1.56.2

View the [commits](#) for this version.

**Release date:** 2024-08-26

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.56.1

View the [commits](#) for this version.

**Release date:** 2024-08-23

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.56.0

View the [commits](#) for this version.

**Release date:** 2024-08-21

This release contains node updates, security and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## Contributors

[CodeShakingSheep](#)

[Oz Weiss](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.55.3

View the [commits](#) for this version.

**Release date:** 2024-08-16

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.55.2

View the [commits](#) for this version.

**Release date:** 2024-08-16

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.55.1

View the [commits](#) for this version.

**Release date:** 2024-08-15

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.54.4

View the [commits](#) for this version.

**Release date:** 2024-08-15

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.54.3

View the [commits](#) for this version.

**Release date:** 2024-08-15

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.54.2

View the [commits](#) for this version.

**Release date:** 2024-08-14

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.55.0

View the [commits](#) for this version.

**Release date:** 2024-08-14

This release contains a new feature, a new node, a node update and bug fixes.

### Override the npm registry

This release adds the option to override the npm registry for installing community packages. This is a paid feature.

We now also prevent npm downloading community packages from a compromised npm registry by explicitly using `-registry` in all npm install commands.

### New node: AI Transform

This release adds the [AI Transform](#) node. Use the AI Transform node to generate code snippets based on your prompt. The AI is context-aware, understanding the workflow's nodes and their data types. The node is only available on [Cloud plans](#).

### New node: Okta

This release adds the [Okta node](#). Use the Okta node to automate work in Okta and integrate Okta with other applications. n8n has built-in support for a wide range of Okta features, which includes creating, updating, and deleting users.

### Node updates

Enhanced node:

- [MySQL](#)

This release also adds the new schema view for the expression editor modal.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.54.1

View the [commits](#) for this version.

**Release date:** 2024-08-13

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.53.2

View the [commits](#) for this version.

**Release date:** 2024-08-08

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.54.0

View the [commits](#) for this version.

**Release date:** 2024-08-07

This release contains new features, node enhancements, bug fixes and updates to our API.

### API update

Our [public REST API](#) now supports additional operations:

- Create, delete, and edit roles for users
- Create, read, update and delete projects

Find the details in the [API reference](#).

## Contributors

[CodeShakingSheep](#)  
[Javier Ferrer González](#)  
[Mickaël Andrieu](#)  
[Oz Weiss](#)  
[Pemontto](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.45.2

View the [commits](#) for this version.

**Release date:** 2024-08-06

This release contains a bug fix.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.53.1

View the [commits](#) for this version.

**Release date:** 2024-08-02

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.53.0

View the [commits](#) for this version.

**Release date:** 2024-07-31

This release contains new features, new nodes, node enhancements, bug fixes and updates to our API.

### **Added Google Cloud Platform Secrets Manager support**

This release adds [Google Cloud Platform Secrets Manager](#) to the list of external secret stores. We already support AWS secrets, Azure Key Vault, Infisical and HashiCorp Vault. External secret stores are available under an enterprise license.

### **New node: Information Extractor**

This release adds the [Information Extractor node](#). The node is specifically tailored for information extraction tasks. It uses Structured Output Parser under the hood, but provides a simpler way to extract information from text in a structured JSON form.

### **New node: Sentiment Analysis**

This release adds the [Sentiment Analysis node](#). The node leverages LLMs to analyze and categorize the sentiment of input text. Users can easily integrate this node into their workflows to perform sentiment analysis on text data. The node is flexible enough to handle various use cases, from basic positive/negative classification to more nuanced sentiment categories.

## Node updates

Enhanced nodes:

- [Calendly Trigger](#)
- [HTTP Request](#)
- [n8n Form Trigger](#)
- [Shopify](#)

## API update

Our [public REST API](#) now supports additional operations:

- Create, read, and delete for variables
- Filtering workflows by project
- Transferring workflows

Find the details in the [API reference](#).

## Contributors

[feelgood-interface](#)  
[Oz Weiss](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.52.2

View the [commits](#) for this version.

**Release date:** 2024-07-31

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.52.1

View the [commits](#) for this version.

**Release date:** 2024-07-26

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.51.2

View the [commits](#) for this version.

**Release date:** 2024-07-26

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.52.0

View the [commits](#) for this version.

**Release date:** 2024-07-25

This release contains new features, node enhancements and bug fixes.

### Added Azure Key Vault support

This release adds [Azure Key Vault](#) to the list of external secret stores. We already support AWS secrets, Infisical and HashiCorp Vault and are working on Google Secrets Manager. External secret stores are available under an enterprise license.

### Node updates

Enhanced nodes:

- [Pinecone Vector Store](#)
- [Supabase Vector Store](#)
- [Send Email](#)

Deprecated nodes:

- OpenAI Model: You can use the OpenAI Chat Model instead
- Google Palm Chat Model: You can use Google Vertex or Gemini instead
- Google Palm Model: You can use Google Vertex or Gemini instead

## n8n@1.51.1

View the [commits](#) for this version.

**Release date:** 2024-07-23

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.50.2

View the [commits](#) for this version.

**Release date:** 2024-07-23

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.51.0

View the [commits](#) for this version.

**Release date:** 2024-07-18

This release contains new nodes, node enhancements and bug fixes.

#### **New node: Text Classifier**

This release adds the [Text Classifier node](#).

#### **New node: Postgres Chat Memory**

This release adds the [Postgres Chat Memory node](#).

#### **New node: Google Vertex Chat Model**

This release adds the [Google Vertex Chat Model node](#).

For full release details, refer to [Releases](#) on GitHub.

### **Node updates**

- Enhanced nodes: Asana

## **n8n@1.50.1**

View the [commits](#) for this version.

**Release date:** 2024-07-16

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## **n8n@1.50.0**

View the [commits](#) for this version.

**Release date:** 2024-07-10

This release contains node enhancements and bug fixes.

### **Node updates**

- Enhanced nodes: Chat Trigger, Google Cloud Firestore, Qdrant Vector Store, Splunk, Telegram
- Deprecated node: Orbit (product shut down)

### **Beta Feature Removal**

The Ask AI beta feature for the HTTP Request node has been removed from this version

### **Contributors**

[Stanley Yoshinori Takamatsu](#)

[CodeShakingSheep](#)

[jeanpaul](#)

[adrian-martinez-onestic](#)

[Malki Davis](#)

## n8n@1.49.0

View the [commits](#) for this version.

**Release date:** 2024-07-03

This release contains a new node, node enhancements, and bug fixes.

### Node updates

- New node added: [Vector Store Tool](#) for the AI Agent
- Enhanced nodes: Zep Cloud Memory, Copper, Embeddings Cohere, GitHub, Merge, Zammad

For full release details, refer to [Releases](#) on GitHub.

### Contributors

[Jochen](#)

[KhDu](#)

[Nico Weichbrodt](#)

[Pavlo Paliychuk](#)

## n8n@1.48.3

View the [commits](#) for this version.

**Release date:** 2024-07-03

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.47.3

View the [commits](#) for this version.

**Release date:** 2024-07-03

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.48.2

View the [commits](#) for this version.

**Release date:** 2024-07-01

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.47.2

View the [commits](#) for this version.

**Release date:** 2024-07-01

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.48.1

View the [commits](#) for this version.

**Release date:** 2024-06-27

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.48.0

View the [commits](#) for this version.

**Release date:** 2024-06-27

This release contains bug fixes and feature enhancements.

For full release details, refer to [Releases](#) on GitHub.

## Contributors

[KubeAl](#)

## n8n@1.47.1

View the [commits](#) for this version.

**Release date:** 2024-06-26

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.47.0

View the [commits](#) for this version.

**Release date:** 2024-06-20

This release contains bug fixes, feature enhancements, a new node, node enhancements and performance improvements.

For full release details, refer to [Releases](#) on GitHub.

### New node: HTTP request tool

This release adds the HTTP request tool. You can use it with an AI agent as a tool to collect information from a website or API. Refer to the [HTTP request tool](#) for details.

## Contributors

[Daniel](#)  
[ekadin-mtc](#)  
[Eric Francis](#)  
[Josh Sorenson](#)  
Mohammad Alsmadi [Nikolai T. Jensen](#)  
[n8n-ninja](#)  
[pebosi](#)  
[Taylor Hoffmann](#)

## n8n@1.45.1

View the [commits](#) for this version.

**Release date:** 2024-06-12

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.46.0

View the [commits](#) for this version.

**Release date:** 2024-06-12

This release contains feature enhancements, node enhancements, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## Contributors

[Jean Khawand](#)  
[pemonto](#)  
[Valentin Coppin](#)

## n8n@1.44.2

View the [commits](#) for this version.

**Release date:** 2024-06-12

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.42.2

View the [commits](#) for this version.

**Release date:** 2024-06-10

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.45.0

View the [commits](#) for this version.

**Release date:** 2024-06-06

This release contains new features, node enhancements, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.44.1

View the [commits](#) for this version.

**Release date:** 2024-06-03

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.44.0

View the [commits](#) for this version.

**Release date:** 2024-05-30

This release contains new features, node enhancements, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.43.1

View the [commits](#) for this version.

**Release date:** 2024-05-28

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.43.0

View the [commits](#) for this version.

**Release date:** 2024-05-22

This release contains new features, node enhancements, and bug fixes.

/// Note | Credential sharing required for manual executions Instance owners and admins: you will see changes if you try to manually execute a workflow where the credentials aren't shared with you. Manual workflow executions now use the same permissions checks as production executions, meaning you can't do a manual execution of a workflow if you don't have access to the credentials. Previously, owners and admins could do manual executions without credentials being shared with them. To resolve this, the credential creator needs to [share the credential](#) with you. ///

### New feature: Projects

With projects and roles, you can give your team access to collections of workflows and credentials, rather than having to share each workflow and credential individually. Simultaneously, you tighten security by limiting access to people on the relevant team.

Refer to the [RBAC](#) documentation for information on creating projects and using roles.

The number of projects and role types vary depending on your plan. Refer to [Pricing](#) for details.

### New node: Slack Trigger

This release adds a trigger node for Slack. Refer to the [Slack Trigger](#) documentation for details.

### Other highlights

- Improved [memory support for OpenAI assistants](#).

### Rolling back to a previous version

If you update to this version, then decide you need to roll back:

Self-hosted n8n:

1. Delete any RBAC projects you created.
2. Revert the database migrations using `n8n db:revert`.

Cloud: contact [help@n8n.io](mailto:help@n8n.io).

### Contributors

[Ayato Hayashi](#)  
[Daniil Zobov](#)  
[Guilherme Barile](#)  
[Romain MARTINEAU](#)

## n8n@1.42.1

View the [commits](#) for this version.

**Release date:** 2024-05-20

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.41.1

View the [commits](#) for this version.

**Release date:** 2024-05-16

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.42.0

View the [commits](#) for this version.

**Release date:** 2024-05-15

This release contains new features, node enhancements, and bug fixes.

Note that this release removes the AI error debugger. We're working on a new and improved version.

### New feature: Tools Agent

This release adds a new option to the Agent node: the [Tools Agent](#).

This agent has an enhanced ability to work with tools, and can ensure a standard output format. This is now the recommended default agent.

For full release details, refer to [Releases](#) on GitHub.

### Contributors

[Mike Quinlan](#)  
[guangwu](#)

## n8n@1.41.0

View the [commits](#) for this version.

**Release date:** 2024-05-08

This release contains new features, node enhancements, and bug fixes.

Note that this release temporarily disables the AI error helper.

For full release details, refer to [Releases](#) on GitHub.

### Contributors

[Florin Lungu](#)

## n8n@1.40.0

View the [commits](#) for this version.

**Release date:** 2024-05-02

This release contains new features, new nodes, node enhancements, and bug fixes.

### New feature: Ask AI in the HTTP node

You can now ask AI to help create API requests in the HTTP Request node:

1. In the HTTP Request node, select **Ask AI**.

2. Enter the **Service** and **Request** you want to use. For example, to use the NASA API to get their picture of the day, enter NASA in **Service** and get picture of the day in **Request**.
3. Check the parameters: the AI tries to fill them out, but you may still need to adjust or correct the configuration.

Self-hosted users need to [enable AI features and provide their own API keys](#)

#### New node: Groq Chat Model

This release adds the [Groq Chat Model node](#).

For full release details, refer to [Releases](#) on GitHub.

#### Contributors

[Alberto Pasqualetto](#)

[Bram Kn](#)

[CodeShakingSheep](#)

[Nicolas-nwb](#)

[pemontto](#)

[pengqiseven](#)

[webk](#)

[Yoshino-s](#)

## n8n@1.39.1

View the [commits](#) for this version.

**Release date:** 2024-04-25

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.38.2

View the [commits](#) for this version.

**Release date:** 2024-04-25

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.37.4

View the [commits](#) for this version.

**Release date:** 2024-04-25

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.39.0

View the [commits](#) for this version.

**Release date:** 2024-04-24

This release contains new nodes, node enhancements, and bug fixes.

#### New node: WhatsApp Trigger

This release adds the [WhatsApp Trigger node](#).

#### Node enhancement: Multiple methods, one Webhook node

The Webhook Trigger node can now handle calls to multiple HTTP methods. Refer to the [Webhook node documentation](#) for information on enabling this.

For full release details, refer to [Releases](#) on GitHub.

### Contributors

[Bram Kn](#)

## n8n@1.38.1

View the [commits](#) for this version.

**Release date:** 2024-04-18

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.37.3

View the [commits](#) for this version.

**Release date:** 2024-04-18

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.38.0

View the [commits](#) for this version.

**Release date:** 2024-04-17

This release contains new nodes, bug fixes, and node enhancements.

#### New node: Google Gemini Chat Model

This release adds the [Google Gemini Chat Model sub-node](#).

#### New node: Embeddings Google Gemini

This release adds the [Google Gemini Embeddings sub-node](#).

For full release details, refer to [Releases](#) on GitHub.

## Contributors

[Chengyou Liu](#)  
[Francesco Mannino](#)

## n8n@1.37.2

View the [commits](#) for this version.

**Release date:** 2024-04-17

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.36.4

View the [commits](#) for this version.

**Release date:** 2024-04-15

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.36.3

View the [commits](#) for this version.

**Release date:** 2024-04-12

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.37.1

View the [commits](#) for this version.

**Release date:** 2024-04-11

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.36.2

View the [commits](#) for this version.

**Release date:** 2024-04-11

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.37.0

[View the commits](#) for this version.

**Release date:** 2024-04-10

This release contains a new node, improvements to error handling and messaging, node enhancements, and bug fixes.

#### **New node: JWT**

This release adds the [JWT core node](#).

For full release details, refer to [Releases](#) on GitHub.

### **Contributors**

[Miguel Prytoluk](#)

## **n8n@1.36.1**

[View the commits](#) for this version.

**Release date:** 2024-04-04

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## **n8n@1.36.0**

[View the commits](#) for this version.

**Release date:** 2024-04-03

This release contains new nodes, enhancements and bug fixes.

#### **New node: Salesforce Trigger node**

This release adds the [Salesforce Trigger node](#).

#### **New node: Twilio Trigger node**

This release adds the [Twilio Trigger node](#).

For full release details, refer to [Releases](#) on GitHub.

## **n8n@1.35.0**

[View the commits](#) for this version.

**Release date:** 2024-03-28

This release contains enhancements and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## **n8n@1.34.2**

[View the commits](#) for this version.

**Release date:** 2024-03-26

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.34.1

[View the commits](#) for this version.

**Release date:** 2024-03-25

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.34.0

[View the commits](#) for this version.

**Release date:** 2024-03-20

This release contains new features, new nodes, and bug fixes.

### New node: Microsoft OneDrive Trigger node

This release adds the [Microsoft OneDrive Trigger node](#). You can now trigger workflows on file and folder creation and update events.

### New data transformation functions

This release introduces new [data transformation functions](#):

#### String

```
toDateTime() //replaces toDate(). toDate() is retained for backwards compatibility.
parseJson()
extractUrlPath()
toBoolean()
base64Encode()
base64Decode()
```

#### Number

```
toDateTime()
toBoolean()
```

#### Object

```
toJsonString()
```

#### Array

```
toJsonString()
```

#### Date & DateTime

```
toDateTime()
toInt()
```

## **Boolean**

[toInt\(\)](#)

## **Contributors**

[Bram Kn](#)  
[pemonto](#)

## **n8n@1.33.1**

View the [commits](#) for this version.

**Release date:** 2024-03-15

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## **n8n@1.32.2**

View the [commits](#) for this version.

**Release date:** 2024-03-15

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## **n8n@1.33.0**

View the [commits](#) for this version.

**Release date:** 2024-03-13

This release contains new features, node enhancements, and bug fixes.

## **Support for Claude 3**

This release adds support for Claude 3 to the [Anthropic Chat Model](#) node.

For full release details, refer to [Releases](#) on GitHub.

## **Contributors**

[gumida](#)  
[Ayato Hayashi](#)  
[Jordan](#)  
[MC Naveen](#)

## **n8n@1.32.1**

View the [commits](#) for this version.

**Release date:** 2024-03-07

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.31.2

View the [commits](#) for this version.

**Release date:** 2024-03-07

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.32.0

View the [commits](#) for this version.

**Release date:** 2024-03-06

This release contains new features, node enhancements, performance improvements, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.31.1

View the [commits](#) for this version.

**Release date:** 2024-03-06

This is a bug fix release and it contains a breaking change.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.31.0

View the [commits](#) for this version.

**Release date:** 2024-02-28

This release contains new features, new nodes, node enhancements and bug fixes.

### New nodes: Microsoft Outlook trigger and Ollama embeddings

This release adds two new nodes.

- [Microsoft Outlook Trigger](#)
- [Ollama Embeddings](#)

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.30.1

View the [commits](#) for this version.

**Release date:** 2024-02-23

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.30.0

View the [commits](#) for this version.

**Release date:** 2024-02-21

This release contains new features, node enhancements, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.29.1

View the [commits](#) for this version.

**Release date:** 2024-02-16

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.29.0

View the [commits](#) for this version.

**Release date:** 2024-02-15

This release contains new features, node enhancements, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

### New features

#### OpenAI node overhaul

This release includes a new version of the [OpenAI node](#), adding more operations, including support for working with assistants.

Other highlights:

- Support for AI events in [log streaming](#).
- Added support for workflow tags in the [public API](#).

### Contributors

[Bruno Inec](#)  
[Jesús Burgers](#)

## n8n@1.27.3

View the [commits](#) for this version.

**Release date:** 2024-02-15

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.28.0

View the [commits](#) for this version.

**Release date:** 2024-02-07

This release contains new features, new nodes, node enhancements and bug fixes.

### New nodes: Azure OpenAI chat model and embeddings

This release adds two new nodes to work with [Azure OpenAI](#) in your advanced AI workflows:

- [Embeddings Azure OpenAI](#)
- [Azure OpenAI Chat Model](#)

For full release details, refer to [Releases](#) on GitHub.

## Contributors

[Andrea Ascari](#)

## n8n@1.27.2

View the [commits](#) for this version.

**Release date:** 2024-02-02

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.27.1

View the [commits](#) for this version.

**Release date:** 2024-01-31

This release contains new features, node enhancements, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.27.0

View the [commits](#) for this version.

**Release date:** 2024-01-31

This release contains node enhancements and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.26.0

[View the commits](#) for this version.

**Release date:** 2024-01-24

This release contains new features, node enhancements, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## Contributors

[Daniel Schröder](#)  
[Nihaal Sangha](#)

## n8n@1.25.1

[View the commits](#) for this version.

**Release date:** 2024-01-22

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## Contributors

[Nihaal Sangha](#)

## n8n@1.25.0

[View the commits](#) for this version.

**Release date:** 2024-01-17

This release contains a new node, feature improvements, and bug fixes.

### New node: Chat Memory Manager

The [Chat Memory Manager](#) node replaces the Chat Messages Retriever node. It manages chat message memories within your AI workflows.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.24.1

[View the commits](#) for this version.

**Release date:** 2024-01-16

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.22.6

View the [commits](#) for this version.

**Release date:** 2024-01-10

This is a bug fix release. It includes important fixes for the HTTP Request and monday.com nodes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.24.0

View the [commits](#) for this version.

**Release date:** 2024-01-10

This release contains new nodes for advanced AI, node enhancements, new features, performance enhancements, and bug fixes.

### Chat trigger

n8n has created a new [Chat Trigger node](#). The new node provides a chat interface that you can make publicly available, with customization and authentication options.

### Mistral Cloud Chat and Embeddings

This release introduces two new nodes to support [Mistral AI](#):

- [Mistral Cloud Chat Model](#)
- [Embeddings Mistral Cloud](#)

### Contributors

[Anush](#)

[Eric Koleda](#)

[Mason Geloso](#)

[vacitbaydarman](#)

## n8n@1.22.5

View the [commits](#) for this version.

**Release date:** 2024-01-09

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.23.0

View the [commits](#) for this version.

**Release date:** 2024-01-03

This release contains new nodes, node enhancements, new features, and bug fixes.

### New nodes and improved experience for working with files

This release includes a major overhaul of nodes relating to files (binary data).

There are now three key nodes dedicated to handling binary data files:

- [Read/Write Files from Disk](#) to read and write files from/to the machine where n8n is running.
- [Convert to File](#) to take input data and output it as a file.
- [Extract From File](#) to get data from a binary format and convert it to JSON.

n8n has moved support for iCalendar, PDF, and spreadsheet formats into these nodes, and removed the iCalendar, Read PDF, and Spreadsheet File nodes. There are still standalone nodes for [HTML](#) and [XML](#).

### New node: Qdrant vector store

This release adds support for [Qdrant](#) with the Qdrant vector store node.

Read n8n's [Qdrant vector store node documentation](#)

### Contributors

[Aaron Gutierrez](#)

[Advaith Gundu](#)

[Anush](#)

[Bin](#)

[Nihaal Sangha](#)

## n8n@1.22.4

View the [commits](#) for this version.

**Release date:** 2024-01-03

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.22.3

View the [commits](#) for this version.

**Release date:** 2023-12-27

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.22.2

View the [commits](#) for this version.

**Release date:** 2023-12-27

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.22.1

View the [commits](#) for this version.

**Release date:** 2023-12-21

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.22.0

View the [commits](#) for this version.

**Release date:** 2023-12-21

This release contains node enhancements, new features, performance improvements, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.18.4

View the [commits](#) for this version.

**Release date:** 2023-12-19

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.21.1

View the [commits](#) for this version.

**Release date:** 2023-12-15

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.18.3

View the [commits](#) for this version.

**Release date:** 2023-12-15

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.21.0

View the [commits](#) for this version.

**Release date:** 2023-12-13

This release contains new features and nodes, node enhancements, and bug fixes.

### New user role: Admin

This release introduces a third account type: admin. This role is available on pro and enterprise plans. Admins have similar permissions to instance owners.

[Read more about user roles](#)

### New data transformation nodes

This release replaces the Item Lists node with a collection of nodes for data transformation tasks:

- Aggregate: take separate items, or portions of them, and group them together into individual items.
- Limit: remove items beyond a defined maximum number.
- Remove Duplicates: identify and delete items that are identical across all fields or a subset of fields.
- Sort: organize lists of in a desired ordering, or generate a random selection.
- Split Out: separate a single data item containing a list into multiple items.
- Summarize: aggregate items together, in a manner similar to Excel pivot tables.

### Increased sharing permissions for owners and admins

Instance owners and users with the admin role can now see and share all workflows and credentials. They can't view sensitive credential information.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.20.0

View the [commits](#) for this version.

**Release date:** 2023-12-06

This release contains bug fixes, node enhancements, and ongoing new feature work.

For full release details, refer to [Releases](#) on GitHub.

### Contributors

[Andrey Starostin](#)

## n8n@1.19.5

View the [commits](#) for this version.

**Release date:** 2023-12-05

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.18.2

View the [commits](#) for this version.

**Release date:** 2023-12-05

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.19.4

View the [commits](#) for this version.

**Release date:** 2023-12-01

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.19.0

View the [commits](#) for this version.

**Release date:** 2023-11-29

This release contains new features, node enhancements, and bug fixes.

### LangChain general availability

This release adds LangChain support to the main n8n version. Refer to [LangChain](#) for more information on how to build AI tools in n8n, the new nodes n8n has introduced, and related learning resources.

### Show avatars of users working on the same workflow

This release improves the experience of users collaborating on workflows. You can now see who else is editing at the same time as you.

## n8n@1.18.1

View the [commits](#) for this version.

**Release date:** 2023-11-30

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.18.0

View the [commits](#) for this version.

**Release date:** 2023-11-22

This release contains new features and bug fixes.

## **Template creator hub**

Built a template you want to share? This release introduces the n8n Creator hub. Refer to the [creator hub Notion doc](#) for more information on this project.

### **Node input and output search filter**

Cloud Pro and Enterprise users can now search and filter the input and output data in nodes. Refer to [Data filtering](#) for more information.

For full release details, refer to [Releases](#) on GitHub.

## **n8n@1.17.1**

View the [commits](#) for this version.

**Release date:** 2023-11-17

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## **n8n@1.17.0**

View the [commits](#) for this version.

**Release date:** 2023-11-15

This release contains node enhancements and bug fixes.

### **Sticky Note Colors**

You can now select background colors for sticky notes.

### **Discord Node Overhaul**

An overhaul of the Discord node, improving the UI making it easier to configure, improving error handling, and fixing issues.

For full release details, refer to [Releases](#) on GitHub.

## **Contributors**

[antondollmaier](#)  
[teomane](#)

## **n8n@1.16.0**

View the [commits](#) for this version.

**Release date:** 2023-11-08

This release contains node enhancements and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.15.2

View the [commits](#) for this version.

**Release date:** 2023-11-07

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.15.1

View the [commits](#) for this version.

**Release date:** 2023-11-02

This release contains new features, node enhancements, and bug fixes.

### Workflow history

This release introduces workflow history: view and load previous versions of your workflows.

Workflow history is available in Enterprise n8n, and with limited history for Cloud Pro.

Learn more in the [Workflow history](#) documentation.

### Dark mode

*Almost* in time for Halloween: this release introduces dark mode.

To enable dark mode:

1. Select **Settings > Personal**.
2. Under **Personalisation**, change **Theme** to **Dark theme**.

### Optional error output for nodes

All nodes apart from sub-nodes and trigger nodes have a new optional output: **Error**. Use this to add steps to handle node errors.

### Pagination support added to HTTP Request node

The HTTP Request node now supports an pagination. Read the [node docs](#) for information and examples.

For full release details, refer to [Releases](#) on GitHub.

### Contributors

[Yoshino-s](#)

## n8n@1.14.2

View the [commits](#) for this version.

**Release date:** 2023-10-26

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.14.1

View the [commits](#) for this version.

**Release date:** 2023-10-26

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.14.0

View the [commits](#) for this version.

**Release date:** 2023-10-25

This release contains node enhancements and bug fixes.

### Switch node supports more outputs

The [Switch node](#) now supports an unlimited number of outputs.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.13.0

View the [commits](#) for this version.

**Release date:** 2023-10-25

This release contains new features, feature enhancements, and bug fixes.

### RSS Feed Trigger node

This releases introduces a new node, the [RSS Feed Trigger](#). Use this node to start a workflow when a new RSS feed item is published.

### Facebook Lead Ads Trigger node

This releases add another new node, the [Facebook Lead Ads Trigger](#). Use this node to trigger a workflow when you get a new lead.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.12.2

View the [commits](#) for this version.

**Release date:** 2023-10-24

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## Contributors

[Burak Akgün](#)

## n8n@1.12.1

View the [commits](#) for this version.

**Release date:** 2023-10-23

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## Contributors

[Léo Martinez](#)

## n8n@1.11.2

View the [commits](#) for this version.

**Release date:** 2023-10-23

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## Contributors

[Inga](#)  
[pemontto](#)

## n8n@1.12.0

View the [commits](#) for this version.

**Release date:** 2023-10-18

This release contains new features, node enhancements, and bug fixes.

### Form Trigger node

This releases introduces a new node, the [n8n Form Trigger](#). Use this node to start a workflow based on a user submitting a form. It provides a configurable form interface.

For full release details, refer to [Releases](#) on GitHub.

## Contributors

[Damian Karzon](#)  
[Inga](#)  
[pemontto](#)

## n8n@1.11.1

View the [commits](#) for this version.

**Release date:** 2023-10-13

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.11.0

View the [commits](#) for this version.

**Release date:** 2023-10-11

This release contains new features and bug fixes.

### External storage for binary files

Self-hosted users can now use an external service to store binary data.  
Learn more in [External storage](#).

If you're using n8n Cloud and are interested in this feature, please [contact n8n](#).

### Item Lists node supports binary data

The Item Lists node now supports splitting and concatenating binary data inputs. This means you no longer need to use code to split a collection of files into multiple items.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.10.1

View the [commits](#) for this version.

**Release date:** 2023-10-11

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.9.3

View the [commits](#) for this version.

**Release date:** 2023-10-10

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.9.2

View the [commits](#) for this version.

**Release date:** 2023-10-09

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.10.0

View the [commits](#) for this version.

**Release date:** 2023-10-05

This release contains bug fixes and preparatory work for new features.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.9.1

View the [commits](#) for this version.

**Release date:** 2023-10-04

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## LangChain in n8n (beta)

**Release date:** 2023-10-04

This release introduces support for building with LangChain in n8n.

With n8n's LangChain nodes you can build AI-powered functionality within your workflows. The LangChain nodes are configurable, meaning you can choose your preferred agent, LLM, memory, and other components. Alongside the LangChain nodes, you can connect any n8n node as normal: this means you can integrate your LangChain logic with other data sources and services.

Read more:

- This is a beta release, and not yet available in the main product. Follow the instructions in [Access LangChain in n8n](#) to try it out. Self-hosted and Cloud options are available.
- Learn how LangChain concepts map to n8n nodes in [LangChain concepts in n8n](#).
- Browse n8n's new [Cluster nodes](#). This is a new set of node types that allows for multiple nodes to work together to configure each other.

## n8n@1.9.0

View the [commits](#) for this version.

**Release date:** 2023-09-28

This release contains new features, performance improvements, and bug fixes.

### Tournament

This releases replaces RiotTmpl, the templating language used in expressions, with n8n's own templating language, [Tournament](#). You can now use arrow functions in expressions.

## **N8N\_BINARY\_DATA\_TTL and EXECUTIONS\_DATA\_PRUNE\_TIMEOUT removed**

The environment variables N8N\_BINARY\_DATA\_TTL and EXECUTIONS\_DATA\_PRUNE\_TIMEOUT no longer have any effect and can be removed. Instead of relying on a TTL system for binary data, n8n cleans up binary data together with executions during pruning.

For full release details, refer to [Releases](#) on GitHub.

## **n8n@1.8.2**

View the [commits](#) for this version.

**Release date:** 2023-09-25

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## **n8n@1.8.1**

View the [commits](#) for this version.

**Release date:** 2023-09-21

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## **n8n@1.8.0**

View the [commits](#) for this version.

**Release date:** 2023-09-20

This release contains node enhancements and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## **n8n@1.7.1**

View the [commits](#) for this version.

**Release date:** 2023-09-14

This release contains bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## **n8n@1.7.0**

View the [commits](#) for this version.

**Release date:** 2023-09-13

This release contains node enhancements and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## Contributors

[Quang-Linh LE](#)  
[MC Naveen](#)

## n8n@1.6.1

View the [commits](#) for this version.

**Release date:** 2023-09-06

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.6.0

View the [commits](#) for this version.

**Release date:** 2023-09-06

This release contains bug fixes, new features, and node enhancements.

### TheHive 5

This release introduces support for TheHive API version 5. This uses a new node and credentials:

- [TheHive 5 node](#)
- [TheHive 5 Trigger node](#)
- [TheHive 5 credentials](#)

### N8N\_PERSISTED\_BINARY\_DATA\_TTL removed

The environment variables N8N\_PERSISTED\_BINARY\_DATA\_TTL no longer has any effect and can be removed. This legacy flag was originally introduced to support ephemeral executions (see [details](#)), which are no longer supported.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.5.1

View the [commits](#) for this version.

**Release date:** 2023-08-31

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.5.0

View the [commits](#) for this version.

**Release date:** 2023-08-31

This release contains new features, node enhancements, and bug fixes.

## Highlights

### External secrets storage for credentials

Enterprise-tier accounts can now use external secrets vaults to manage credentials in n8n. This allows you to store credential information securely outside your n8n instance. n8n supports Infisical and HashiCorp Vault.

Refer to [External secrets](#) for guidance on enabling and using this feature.

### Two-factor authentication

n8n now supports two-factor authentication (2FA) for self-hosted instances. n8n is working on bringing support to Cloud. Refer to [Two-factor authentication](#) for guidance on enabling and using it.

### Debug executions

Users on a paid n8n plan can now load data from previous executions into their current workflow. This is useful when debugging a failed execution.

Refer to [Debug executions](#) for guidance on using this feature.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.4.1

View the [commits](#) for this version.

**Release date:** 2023-08-29

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.4.0

View the [commits](#) for this version.

**Release date:** 2023-08-23

This release contains new features, node enhancements, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## Contributors

[pemontto](#)

## n8n@1.3.1

[View the commits](#) for this version.

**Release date:** 2023-08-18

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.3.0

[View the commits](#) for this version.

**Release date:** 2023-08-16

This release contains new features and bug fixes.

### Highlights

#### Trial feature: AI support in the Code node

This release introduces limited support for using AI to generate code in the Code node. Initially this feature is only available on Cloud, and will gradually be rolled out, starting with about 20% of users.

Learn how to use the feature, including guidance on writing prompts, in [Generate code with ChatGPT](#).

For full release details, refer to [Releases](#) on GitHub.

### Contributors

[Ian Gallagher](#)

[Xavier Calland](#)

## n8n@1.2.2

[View the commits](#) for this version.

**Release date:** 2023-08-14

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.2.1

[View the commits](#) for this version.

**Release date:** 2023-08-09

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.2.0

[View the commits](#) for this version.

**Release date:** 2023-08-09

This release contains new features, node enhancements, bug fixes, and performance improvements.

## Highlights

### Credential support for SecOps services

This release introduces support for setting up credentials in n8n for the following services:

- [AlienVault](#)
- [Auth0 Management](#)
- [Carbon Black API](#)
- [Cisco Meraki API](#)
- [Cisco Secure Endpoint](#)
- [Cisco Umbrella API](#)
- [CrowdStrike](#)
- [F5 Big-IP](#)
- [Fortinet FortiGate](#)
- [Hybrid Analysis](#)
- [Imperva WAF](#)
- [Kibana](#)
- [Microsoft Entra ID](#)
- [Mist](#)
- [Okta](#)
- [OpenCTI](#)
- [QRadar](#)
- [Qualys](#)
- [Recorded Future](#)
- [Sekoia](#)
- [Shuffler](#)
- [Trellix ePO](#)
- [VirusTotal](#)
- [Zscaler ZIA](#)

This makes it easier to do [Custom operations](#) with these services, using the [HTTP Request](#) node.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.1.1

View the [commits](#) for this version.

**Release date:** 2023-07-27

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.1.0

View the [commits](#) for this version.

**Release date:** 2023-07-26

This release contains new features, bug fixes, and node enhancements.

## Highlights

### Source control and environments

This release introduces source control and environments for enterprise users.

n8n uses Git-based source control to support environments. Linking your n8n instances to a Git repository lets you create multiple n8n environments, backed by Git branches.

Refer to [Source control and environments](#) to learn more about the features and set up your environments.

For full release details, refer to [Releases](#) on GitHub.

### Contributors

[Adrián Martínez](#)

[Alberto Pasqualetto](#)

[Marten Steketee](#)

[perseus-algo1](#)

[Sandra Ashipala](#)

[ZergRael](#)

## n8n@1.0.5

View the [commits](#) for this version.

**Release date:** 2023-07-24

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@1.0.4

View the [commits](#) for this version.

**Release date:** 2023-07-19

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

### Contributors

[Romain Dunand](#)

[noctarius aka Christoph Engelbert](#)

## n8n@1.0.3

View the [commits](#) for this version.

**Release date:** 2023-07-13

This release contains API enhancements and adds support for sending messages to forum threads in the Telegram node.

For full release details, refer to [Releases](#) on GitHub.

## Contributors

[Kirill](#)

## n8n@1.0.2

View the [commits](#) for this version.

**Release date:** 2023-07-05

This is a bug fix release.

## Contributors

[Romain Dunand](#)

## n8n@1.0.1

View the [commits](#) for this version.

**Release date:** 2023-07-05

This is n8n's version one release.

For full details, refer to the [n8n v1.0 migration guide](#).

## Highlights

### Python support

Although JavaScript remains the default language, you can now also select Python as an option in the [Code node](#) and even make use of [many Python modules](#). Note that Python is unavailable in Code nodes added to a workflow before v1.0.

## Contributors

[Marten Steketee](#)

---

## Release notes pre 1.0

Features and bug fixes for n8n before the release of 1.0.0.

You can also view the [Releases](#) in the GitHub repository.

-8<- “\_snippets/self-hosting/installation/latest-next-version.md”

-8<- “\_snippets/update-n8n.md”

## Semantic versioning in n8n

n8n uses [semantic versioning](#). All version numbers are in the format MAJOR.MINOR.PATCH. Version numbers increment as follows:

- MAJOR version when making incompatible changes which can require user action.
- MINOR version when adding functionality in a backward-compatible manner.
- PATCH version when making backward-compatible bug fixes.

## n8n@0.237.0

[View the commits](#) for this version.

**Release date:** 2023-08-17

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

### Contributors

[Jordan Hall](#)

[Xavier Calland](#)

## n8n@0.236.3

[View the commits](#) for this version.

**Release date:** 2023-07-18

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

### Contributors

[Romain Dunand](#)

[noctarius aka Christoph Engelbert](#)

## n8n@0.236.2

[View the commits](#) for this version.

**Release date:** 2023-07-14

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@0.236.1

[View the commits](#) for this version.

**Release date:** 2023-07-12

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@0.236.0

View the [commits](#) for this version.

**Release date:** 2023-07-05

This release contains new nodes, node enhancements, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

### New nodes

#### **crowd.dev**

This release includes a [crowd.dev](#) node and crowd.dev Trigger node. crowd.dev is a tool to help you understand who is engaging with your open source project.

[crowd.dev node documentation](#).

### Contributors

[Alberto Pasqualetto](#)  
[perseus-algol](#)  
[Romeo Balta](#)  
[ZergRael](#)

## n8n@0.234.1

View the [commits](#) for this version.

**Release date:** 2023-07-05

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@0.235.0

View the [commits](#) for this version.

**Release date:** 2023-06-28

This release contains new features, new nodes, node enhancements, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

### Contributors

[Marten Steketee](#)  
[Sandra Ashipala](#)

## n8n@0.234.0

View the [commits](#) for this version.

**Release date:** 2023-06-22

This release contains new features, new nodes, node enhancements, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## New nodes

### Debug Helper

The Debug Helper node can be used to trigger different error types or generate random datasets to help test n8n workflows.

[Debug Helper node documentation](#).

## n8n@0.233.1

View the [commits](#) for this version.

**Release date:** 2023-06-19

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@0.233.0

View the [commits](#) for this version.

**Release date:** 2023-06-14

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@0.232.0

View the [commits](#) for this version.

**Release date:** 2023-06-07

This release contains new features, new nodes, node enhancements, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

## New nodes

This release includes a new trigger node for Postgres, which allows you to listen to events, as well as listen to custom channels. Refer to [Postgres Trigger](#) for more information.

## n8n@0.231.3

View the [commits](#) for this version.

**Release date:** 2023-06-17

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@0.231.2

View the [commits](#) for this version.

**Release date:** 2023-06-14

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@0.231.1

View the [commits](#) for this version.

**Release date:** 2023-06-06

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@0.231.0

View the [commits](#) for this version.

**Release date:** 2023-05-31

This release contains bug fixes and new features.

For full release details, refer to [Releases](#) on GitHub.

### New features

Notable new features.

#### Resource mapper UI component

This release includes a new UI component, the resource mapper. This component is useful for node creators. If your node does insert, update, or upsert operations, you need to send data from the node in a format supported by the service you're integrating with. Often it's necessary to use a Set node before a node that sends data, to get the data to match the schema of the service you're connecting to. The resource mapper UI component provides a way to get data into the required format directly within the node.

Refer to [Node user interface elements | Resource mapper](#) for guidance for node builders.

## n8n@0.230.3

View the [commits](#) for this version.

**Release date:** 2023-06-05

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@0.230.2

View the [commits](#) for this version.

**Release date:** 2023-05-25

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@0.230.1

View the [commits](#) for this version.

**Release date:** 2023-05-25

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@0.230.0

View the [commits](#) for this version.

**Release date:** 2023-05-24

This release contains new features, new nodes, node enhancements, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

### New nodes

#### Execution Data

Save metadata for workflow executions. You can then search by this data in the **Executions** list.

[Execution Data node documentation](#).

#### LDAP node

The LDAP node allows you to interact with your LDAP servers from your n8n workflows.

[LDAP node documentation](#).

#### LoneScale node

Integrate n8n with [LoneScale](#), a buying intents data platform.

[LoneScale node documentation](#).

### Contributors

[Bram Knempton](#)  
[Yann Aleman](#)

## n8n@0.229.0

View the [commits](#) for this version.

**Release date:** 2023-05-17

This release contains bug fixes, improves UI copy and error messages in some nodes, and other node enhancements.

For full release details, refer to [Releases](#) on GitHub.

### Node enhancements

The Google Ads node now supports v13.

## n8n@0.228.2

View the [commits](#) for this version.

**Release date:** 2023-05-15

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@0.228.1

View the [commits](#) for this version.

**Release date:** 2023-05-11

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@0.228.0

View the [commits](#) for this version.

**Release date:** 2023-05-11

This release contains new features, node enhancements, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

### New nodes

#### npm node

This release introduces the [npm](#) node. This is a new core node. It provides a way to query an npm registry within your workflow.

### Contributors

[Adam Charnock](#)

## n8n@0.227.1

View the [commits](#) for this version.

**Release date:** 2023-05-15

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@0.227.0

View the [commits](#) for this version.

**Release date:** 2023-05-03

This release contains new features, node enhancements, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

### Node enhancements

- An overhaul of the Microsoft Excel 365 node, improve the UI making it easier to configure, improve error handling, and fix issues.

### Deprecations

This release deprecates the following:

- The EXECUTIONS\_PROCESS environment variable.
- Running n8n in own mode. Main mode is now the default. Use [Queue mode](#) if you need full execution isolation.
- The WEBHOOK\_TUNNEL\_URL flag. Replaced by WEBHOOK\_URL.
- Support for MySQL and MariaDB as n8n backend databases. n8n will remove support completely in version 1.0. n8n recommends using PostgreSQL instead.

## n8n@0.226.2

View the [commits](#) for this version.

**Release date:** 2023-05-03

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@0.226.1

View the [commits](#) for this version.

**Release date:** 2023-05-02

This is a bug fix release.

For full release details, refer to [Releases](#) on GitHub.

## n8n@0.226.0

View the [commits](#) for this version.

**Release date:** 2023-04-26

This release contains new features, node enhancements, and bug fixes.

For full release details, refer to [Releases](#) on GitHub.

### New features

- A new command to get information about licenses for self-hosted users: sh n8n license:info

### Node enhancements

- Nodes that use SQL, such as the PostgreSQL node, now have a better SQL editor for writing custom queries.
- An overhaul of the Google BigQuery node to support executing queries, improve the UI making it easier to configure, improve error handling, and fix issues.

## n8n@0.225.2

View the [commits](#) for this version.

**Release date:** 2023-04-25

This is a bug fix release.

### Bug fixes

- Core: Upgrade google-timezones-json to use the correct timezone for Sao Paulo.
- Code Node: Update vm2 to address [CVE-2023-30547](#).

## n8n@0.225.1

View the [commits](#) for this version.

**Release date:** 2023-04-20

This is a bug fix release.

### Bug fixes

- Editor: Clean up demo and template callouts from workflows page.
- Editor: Fix memory leak in Node Detail View by correctly unsubscribing from event buses.
- Editor: Settings sidebar should disconnect from push when navigating away.
- Notion Node: Update credential test to not require user permissions.

## n8n@0.225.0

[View the commits](#) for this version.

**Release date:** 2023-04-19

## New features

This release introduces [Variables](#). You can now create variables that allows you to store and reuse values in n8n workflows. This is the first phase of a larger project to support [Environments](#) in n8n.

- Core: Add support for Google Service account authentication in the HTTP Request node.
- GitLab Node: Add **Additional Parameters** for the file list operation.
- MySQL Node: This node has been overhauled.

## Bug fixes

- Core: Fix broken API permissions in public API.
- Core: Fix paired item returning wrong data.
- Core: Improve SAML connection test result views.
- Core: Make getExecutionId available on all nodes types.
- Core: Skip SAML onboarding for users with first- and lastname.
- Editor: Add padding to prepend input.
- Editor: Clean up demo/video experiment.
- Editor: Enterprise features missing with user management.
- Editor: Fix moving canvas on middle click preventing lasso selection.
- Editor: Make sure to redirect to blank canvas after personalisation modal.
- Editor: Fix an issue that was preventing typing certain characters in the UI on devices with touchscreen.
- Editor: Fix n8n-checkbox alignment.
- Code Node: Handle user code returning null and undefined.
- GitHub Trigger Node: Remove content\_reference event.
- Google Sheets Trigger Node: Return actual error message.
- HTTP Request Node: Fix itemIndex in HTTP Request errors.
- NocoDB Node: Fix for updating or deleting rows with not default primary keys.
- OpenAI Node: Update models to only show those supported.
- OpenAI Node: Update OpenAI Text Moderate input placeholder text.

## Contributors

[Bram Kn](#)

[Eddy Hernandez](#)

[Filipe Dobreira](#)

[Jimw383](#)

## n8n@0.224.4

[View the commits](#) for this version.

**Release date:** 2023-04-24

This is a bug fix release.

## Bug fixes

- Core: Upgrade google-timezones-json to use the correct timezone for Sao Paulo.
- Code Node: Update vm2 to address [CVE-2023-30547](#).

## n8n@0.224.2

View the [commits](#) for this version.

**Release date:** 2023-04-20

This is a bug fix release.

### Bug fixes

- Core: Fix paired item returning wrong data.
- Core: Make getExecutionId available on all nodes types.
- Editor: Fix memory leak in Node Detail View by correctly unsubscribing from event buses.
- Editor: Fix moving canvas on middle click preventing lasso selection.
- Editor: Settings sidebar should disconnect from push when navigating away.
- Google Sheets Trigger Node: Return actual error message.
- HTTP Request Node: Fix itemIndex in HTTP Request errors.
- Notion Node: Update credential test to not require user permissions.

### Contributors

[Filipe Dobreira](#)

## n8n@0.224.1

View the [commits](#) for this version.

**Release date:** 2023-04-14

This is a bug fix release.

### Bug fixes

- Core: Fix broken API permissions in public API.
- Editor: Fix an issue that was preventing typing certain characters in the UI on devices with touchscreen.

## n8n@0.224.0

View the [commits](#) for this version.

**Release date:** 2023-04-12

This release contains a new node, updates, and bug fixes.

### New nodes

This release introduces the [TOTP](#) node. This is a new core node. It provides a way to generate a TOTP (time-based one-time password) within your workflow.

## Bug fixes

- Code Node: Update vm2 to address CVE-2023-29017.
- Core: App shouldn't crash with a custom REST endpoint.
- Core: Do not execute workflowExecuteBefore hook when resuming executions from a waiting state.
- Core: Fix issue where sub workflows would display as running forever after failure to start.
- Core: Update xml2js to address CVE-2023-0842.
- Editor: Drop mergeDeep in favor of lodash merge.
- HTTP Request Node: Restore detailed error message.

## Contributors

[Loganaden Velvindron](#)

## n8n@0.223.0

View the [commits](#) for this version.

**Release date:** 2023-04-05

This release contains new features and bug fixes.

## New features

- Core: Convert eventBus controller to decorator style and improve permissions.
- Core: Prevent non owners password reset when SAML is enabled (this is preparation for an upcoming feature).
- Core: Read ephemeral license from environment and clean up ee flags.
- Editor: Allow tab to accept completion.
- Editor: Enable saving workflow when node details view is open.
- Editor: SSO onboarding (this is preparation for an upcoming feature).
- Editor: SSO setup (this is preparation for an upcoming feature).

## Node enhancements

- Filter Node: Show discarded items.
- HTTP Request Node: Follow redirects by default.
- Postgres Node: Overhaul node.
- ServiceNow Node: Add support for work notes when updating an incident.
- SSH Node: Hide the private key within the SSH credential.

## Bug fixes

- Add droppable state for booleans when mapping.
- Compare Datasets Node: Fuzzy compare'n't comparing keys missing in one of the inputs.
- Compare Datasets Node: Fix support for dot notation in skip fields.

- Core: Deactivate active workflows during import.
- Core: Stop marking duplicates as circular references in `jsonStringify`.
- Core: Stop using `util.types.isProxy` for tracking of augmented objects.
- Core: Fix curl import error when no data.
- Core: Handle Date and RegExp correctly in `jsonStringify`.
- Core: Handle Date and RegExp objects in `augmentObject`.
- Core: Prevent `augmentObject` from creating infinitely deep proxies.
- Core: Service account private key as a password field.
- Core: Update lock file.
- Core: Waiting workflows not stopping.
- Date & Time Node: Add info box at top of date and time explaining expressions.
- Date & Time Node: Convert Luxon DateTime object to ISO.
- Editor: Add `$if`, `$min`, `$max` to root expression autocomplete.
- Editor: Curb overeager item access linting.
- Editor: Disable Grammarly in expression editors.
- Editor: Disable password reset on desktop with no user management.
- Editor: Fix connection lost hover text not showing.
- Editor: Fix issue preventing execution preview loading when in an Iframe.
- Editor: Fix mapping with special characters.
- Editor: Prevent error from showing-up when duplicating unsaved workflow.
- Editor: Prevent NDV schema view pagination.
- Editor: Support backspacing with modifier key.
- Google Sheets Node: Fix `insertOrUpdate` cell update with object.
- HTML Extract Node: Support for dot notation in JSON property.
- HTTP Request Node: Fix AWS credentials to stop removing URL parameters for STS.
- HTTP Request Node: Refresh token properly on never fail option.
- HTTP Request Node: Support for dot notation in JSON body.
- LinkedIn Node: Update the version of the API.
- Redis Node: Fix issue with hash set not working as expected.

## n8n@0.222.3

View the [commits](#) for this version.

**Release date:** 2023-04-14

This is a bug fix release.

### Bug fixes

- Core: Fix broken API permissions in public API.
- Editor: Fix an issue that was preventing typing certain characters in the UI on devices with touchscreen.

## n8n@0.222.2

View the [commits](#) for this version.

**Release date:** 2023-04-11

This is a bug fix release.

## Bug fixes

- Code node: Update vm2 to address CVE-2023-29017.
- Core: Update xml2js to address CVE-2023-0842.

## Contributors

[Loganaden Velvindron](#)

## n8n@0.222.1

View the [commits](#) for this version.

**Release date:** 2023-04-04

This is a bug fix release.

## Bug fixes

- AWS SNS Node: Fix an issue with messages failing to send if they contain certain characters.
- Core: augmentObject should clone Buffer/Uint8Array instead of wrapping them in a proxy.
- Core: augmentObject should use existing property descriptors whenever possible.
- Core: Fix the issue of nodes not loading when run using npx.
- Core: Improve Axios error handling in nodes.
- Core: Password reset should pass in the correct values to external hooks.
- Core: Prevent augmentObject from creating infinitely deep proxies.
- Core: Use table-prefixes in queries in import commands.
- Editor: Fix focused state in Code node editor.
- Editor: Fix loading executions in long execution list.
- Editor: Show correct status on canceled executions.
- Gmail Node: Gmail Luxon object support, fix for timestamp.
- HTTP Request Node: Detect mime-type from streaming responses.
- HubSpot Trigger Node: Developer API key is required for webhooks.
- Set Node: Convert string to number.

## n8n@0.222.0

View the [commits](#) for this version.

**Release date:** 2023-03-30

This release contains new features, including custom filters for the executions list, and a new node to filter items in your workflows.

## New features

This release introduces improvements to the execution lists. You can now save [Custom execution data](#), and use it to filter both the [All executions](#) and [Single workflow executions](#) lists.

- Add test overrides.
- Core: Improve LDAP/SAML toggle and tests.
- Core: Limit user invites when SAML is enabled.

- Core: Make OAuth2 error handling consistent with success handling.
- Editor: Fix ResourceLocator dropdown style.

## New nodes

This release introduces the [Filter](#) node. The node allows you to filter items based on a condition. If the item meets the condition, the Filter node passes it on to the next node in the Filter node output. If the item doesn't meet the condition, the Filter node omits the item from its output.

## Bug fixes

- Core: Assign `properties.success` earlier to set `executionStatus` correctly.
- Core: Don't mark duplicates as circular references in `jsonStringify`.
- Core: Don't use `util.types.isProxy` for tracking of augmented objects.
- Core: Ensure that all non-lazy-loaded community nodes get post-processed correctly.
- Core: Force-upgrade `decode-uri-component` to address CVE-2022-38900.
- Core: Force-upgrade `http-cache-semantics` to address CVE-2022-25881.
- Core: Handle Date and RegExp correctly in `jsonStringify`.
- Core: Handle Date and RegExp objects in `augmentObject`.
- Core: Improve Axios error handling in nodes.
- Core: Improve community nodes loading.
- Core: Initialize queue in the webhook server as well.
- Core: Persist `CurrentAuthenticationMethod` setting change.
- Core: Remove circular references from Code and push message.
- Core: Require authentication on icons and nodes/credentials types static files.
- Core: Return SAML service provider URLs with configuration.
- Core: Service account private key should display as a password field.
- Core: Upgrade Luxon to address CVE-2023-22467.
- Core: Upgrade simple-git to address CVE-2022-25912.
- Core: Upgrade SQLite3 to address CVE-2022-43441.
- Core: Upgrade Convict to address CVE-2023-0163.
- Core: Waiting workflows not stopping.
- Editor: Fix connection lost hover text not showing.
- Editor: Fix issue preventing execution preview loading when in an iframe.
- Editor: Use credentials when fetching node and credential types.
- Google Sheets Node: Fix `insertOrUpdate` cell update with object.
- HTTP Request Node: Add streaming to binary response.
- HTTP Request Node: Fix AWS credentials to automatically deconstruct the URL.
- HTTP Request Node: Fix AWS credentials to stop removing URL parameters for STS.
- Split In Batches Node: Roll back changes in v1 and create v2.
- Update PostHog no-capture.

## Contributors

[Manish Dhanwal](#)

## n8n@0.221.3

View the [commits](#) for this version.

**Release date:** 2023-04-11

This is a bug fix release.

### Bug fixes

- Code node: Update vm2 to address CVE-2023-29017.
- Core: Update xml2js to address CVE-2023-0842.

### Contributors

[Loganaden Velvindron](#)

## n8n@0.221.2

View the [commits](#) for this version.

**Release date:** 2023-03-24

This is a bug fix release. It fixes an issue with `properties.success` that was causing `executionStatus` to sometimes be incorrect.

## n8n@0.221.1

View the [commits](#) for this version.

**Release date:** 2023-03-23

This is a bug fix release. It ensures the job queue is initiated before starting the webhook server.

## n8n@0.221.0

View the [commits](#) for this version.

**Release date:** 2023-03-23

### New features

- Core: n8n now augments data rather than copying it in the Code node. This is a performance improvement.
- Editor: you can now move the canvas by holding Space and dragging with the mouse, or by holding the middle mouse button and dragging.
- Editor: add authentication type recommendations in the credentials modal.
- Editor: add the SSO login button.

### New nodes

This release adds a node for [QuickChart](#), an open source chart generation tool.

## Bug fixes

- Core: ensure n8n calls available error workflows in main mode recovery.
- Core: fix telemetry execution status for manual workflows executions.
- Core: return SAML attributes after connection test.
- Editor: disable mapping tooltip for display modes that don't support mapping.
- Editor: fix execution list item selection.
- Editor: fix for large notifications being cut off.
- Editor: fix redo in code and expression editor.
- Editor: fix the canvas node distance when automatically injecting manual trigger.
- HTTP Request Node: fix AWS credentials to automatically deconstruct the URL.
- Split In Batches Node: roll back changes in v1 and create v2.

## n8n@0.220.1

View the [commits](#) for this version.

**Release date:** 2023-03-22

This is a bug fix release. It reverts changes to version 1 of the Split In Batches node, and creates a version 2 containing the updates.

## n8n@0.220.0

View the [commits](#) for this version.

**Release date:** 2023-03-16

This release adds schema view to the node output panel, and includes node enhancements and bug fixes.

## New features

- Core: improve SAML connection test.
- Editor: add basic Datatable and Pagination components.
- Editor: add support for schema view in the NDV output.
- Editor: don't show actions panel for single-action nodes.

## Node enhancements

- Item Lists Node: update actions text.
- OpenAI Node: add support for GPT4 on chat completion.
- Split In Batches Node: make it easier to combine processed data.

## Bug fixes

- Core: initialize license and LDAP in the correct order.
- Editor: display correct error message for \$env access.
- Editor: fix autocomplete for complex expressions.
- Editor: fix owner set-up checkbox wording.
- Editor: properly handle mapping of dragged expression if it contains hyphen.
- Metabase Node: fix issue with question results not correctly being

returned.

## n8n@0.219.1

View the [commits](#) for this version.

**Release date:** 2023-03-10

This is a bug fix release. It resolves an issue with the HTTP Request node by removing the streaming response.

## n8n@0.219.0

View the [commits](#) for this version.

**Release date:** 2023-03-09

### New features

- Core: add advancedFilters feature flag.
- Core: add SAML post and test endpoints.
- Core: add SAML XML validation.
- Core: limit user changes when SAML is enabled.
- Core: refactor and add SAML preferences for service provider instance.
- Editor: don't automatically add the manual trigger when the user adds another node.
- Editor: redirect users to canvas if they don't have any workflows.

### Node enhancements

- Cal Trigger Node: update to support v2 webhooks.
- HTTP Request Node: move from binary buffer to binary streaming.
- Mattermost Node: add self signed certificate support.
- Microsoft SQL Node: add support for self signed certificates.
- Mindee Node: add support for v4 API.
- Slack Node: move from binary buffer to binary streaming.

### Bug fixes

- Core: allow serving icons for custom nodes with npm scoped names.
- Core: rename advancedFilters to advancedExecutionFilters.
- Editor: fix ElButton overrides.
- Editor: only fetch new versions at app launch.
- Fetch credentials on workflows view to include in duplicated workflows.
- Fix color discrepancies for executions list items.
- OpenAI Node: fix issue with expressions not working with chat complete.
- OpenAI Node: simplify code.

### Contributors

[Syed Ali Shahbaz](#)

## n8n@0.218.0

View the [commits](#) for this version.

**Release date:** 2023-03-02

This release contains node enhancements, bug fixes, and new features that lay groundwork for upcoming releases, along with some UX improvements.

### New features

- Add distribution test tracking.
- Add events to enable onboarding checklist.
- Core: add SAML login setup (for upcoming feature).
- Core: add SAML settings and consolidate LDAP under SSO (for upcoming feature).
- Editor: add missing documentation to autocomplete items for inline code editor.
- Editor: Show parameter hint on multiline inputs.

### Node enhancements

- JIRA node: support binary streaming for very large binary files.
- OpenAI node: add support for ChatGPT.
- Telegram node: add parse mode option to Send Document operation.

### Bug fixes

- Core: fix execution pruning queries.
- Core: fix filtering workflow by tags.
- Core: revert isPending check on the user entity.
- Fix issues with nodes missing in nodes panel.
- Fix mapping paths when appending to empty expression.
- Item Lists Node: tweak item list summarize field naming.
- Prevent executions from displaying as running forever.
- Show Execute Workflow node in the nodes panel.
- Show RabbitMQ node in the nodes panel.
- Stop showing mapping hint after mapping.

## n8n@0.217.2

View the [commits](#) for this version.

**Release date:** 2023-02-27

This is a bug fix release.

### Bug fixes

- Core: fix issue with execution pruning queries.
- Core: fix for workflow filtering by tag.
- Core: revert isPending check on the user entity.

## n8n@0.217.1

View the [commits](#) for this version.

**Release date:** 2023-02-24

This is a bug fix release.

## Bug fixes

Prevent executions appearing to run forever.

# n8n@0.217.0

View the [commits](#) for this version.

**Release date:** 2023-02-23

This release contains new features and bug fixes. It includes improvements to the nodes panel and executions list. It also deprecates the Read Binary File node.

## New features

- Add new event hooks to support telemetry around the new onboarding experience.
- Update nodes to set required path type.
- Core: add configurable execution history limit. Use this to improve performance when self-hosting. Refer to [Execution Data | Enable data pruning](#) for more information.
- Core: add execution runData recovery and status field. This allows us to show execution statuses on the **Executions** list.
- Core: add SAML feature flag. This is preparatory for an upcoming feature.
- Editor: improvements to the nodes panel search. When searching in root view, n8n now displays results from both trigger and regular nodes. When searching in a category view, n8n shows results from the category, and also suggests results from other categories.
- Hide sensitive value in authentication header credentials and authentication query credentials.
- Support feature flag evaluation server side.
- Deprecate the Read Binary File node. Use the [Read Binary Files](#) node instead.

## Bug fixes

- Baserow Node: fix issue with **Get All** not correctly using filters.
- Compare Datasets Node: UI tweaks and fixes.
- Core: don't allow arbitrary path traversal in BinaryDataManager.
- Core: don't allow arbitrary path traversal in the credential-translation endpoint.
- Core: don't explicitly bypass authentication on URLs containing .svg.
- Core: don't remove empty output connections arrays in PurgeInvalidWorkflowConnections migration.
- Core: fix execution status filters.
- Core: user update endpoint should only allow updating email, firstName, and lastName.
- Discord Node: fix wrong error message being displayed.
- Discourse Node: fix issue with credential test not working.

- Editor: apply correct IRunExecutionData to finished workflow.
- Editor: fix an issue with zoom and canvas nodes connections.
- Editor: fix unexpected date rendering on front-end.
- Editor: remove crashed status from filter.
- Fix typo in error messages when a property doesn't exist.
- Fixes an issue when saving an active workflow without triggers would cause n8n to be stuck.
- Google Calendar Node: fix incorrect labels for start and end times when getting all events.
- Postgres Node: fix for tables containing field named JSON.
- AWS S3 Node: fix issue with get many buckets not outputting data.

-8<- “\_snippets/update-n8n.md”

## n8n@0.216.3

View the [commits](#) for this version.

**Release date:** 2023-03-09

This is a bug fix release. It reverts the `isPending` check on the user entity, resolving an issue with displaying user options when user management is disabled.

## n8n@0.216.2

View the [commits](#) for this version.

**Release date:** 2023-02-23

This is a bug fix release.

### Bug fixes

Core: don't remove empty output connections arrays in `PurgeInvalidWorkflowConnections` migration.

## n8n@0.215.4

View the [commits](#) for this version.

**Release date:** 2023-03-14

This is a bug fix release. It reverts the `isPending` check on the user entity, resolving an issue with displaying user options when user management is disabled.

-8<- “\_snippets/update-n8n.md”

## n8n@0.215.3

View the [commits](#) for this version.

**Release date:** 2023-02-23

This is a bug fix release. It contains an important security fix.

### Bug fixes

- Core: don't allow arbitrary path traversal in BinaryDataManager.
- Core: don't allow arbitrary path traversal in the credential-translation endpoint.
- Core: don't explicitly bypass authentication on URLs containing .svg.
- Core: don't remove empty output connections arrays in PurgeInvalidWorkflowConnections migration.
- Core: the user update endpoint should only allow updating email, first name, and last name.

## n8n@0.214.5

View the [commits](#) for this version.

**Release date:** 2023-03-14

This is a bug fix release. It reverts the isPending check on the user entity, resolving an issue with displaying user options when user management is disabled.

-8<- “\_snippets/update-n8n.md”

## n8n@0.214.4

View the [commits](#) for this version.

**Release date:** 2023-02-23

This is a bug fix release. It contains an important security fix.

### Bug fixes

- Core: don't allow arbitrary path traversal in BinaryDataManager.
- Core: don't allow arbitrary path traversal in the credential-translation endpoint.
- Core: don't explicitly bypass authentication on URLs containing .svg.
- Core: don't remove empty output connections arrays in PurgeInvalidWorkflowConnections migration.
- Core: the user update endpoint should only allow updating email, first name, and last name.

## n8n@0.216.1

View the [commits](#) for this version.

**Release date:** 2023-02-21

This is a bug fix release.

### Bug fixes

- Core: don't allow arbitrary path traversal in BinaryDataManager.
- Core: don't allow arbitrary path traversal in the credential-translation endpoint.
- Core: don't explicitly bypass auth on URLs containing .svg.
- Core: user update endpoint should only allow updating email, firstName, and lastName.

## n8n@0.216.0

[View the commits](#) for this version.

**Release date:** 2023-02-16

This release contains new features, node enhancements, and bug fixes.

### New features

- Add workflow and credential sharing access e2e tests.
- Editor: add correct credential owner contact details for readonly credentials.
- Editor: add most important native properties and methods to autocomplete.
- Editor: update to personalization survey v4.
- Update telemetry API endpoints.

### Node enhancements

- GitHub node: update code to use resource locator component.
- GitHub Trigger node: update code to use resource locator component.
- Notion node: add option to set icons when creating pages or database pages.
- Slack node: add support for manually inputting a channel name for channel operations.

### Bug fixes

- Core: fix data transformation functions.
- Core: remove unnecessary info from GET /workflows response.
- Bubble node: fix pagination issue when returning all objects.
- HTTP Request Node: ignore empty body when auto-detecting JSON.

### Contributors

[feelgood-interface](#)

## n8n@0.215.2

[View the commits](#) for this version.

**Release date:** 2023-02-14

This is a bug fix release. It solves an issue that was causing webhooks to be removed when they shouldn't be.

## n8n@0.215.1

[View the commits](#) for this version.

**Release date:** 2023-02-11

This is a bug fix release.

## Bug fixes

- Core: fix issue causing worker and webhook service to close on start.
- Core: handle versioned custom nodes correctly.

## n8n@0.215.0

View the [commits](#) for this version.

**Release date:** 2023-02-10

This release contains new features, node enhancements, and bug fixes.

## New features

- Refactor the n8n Desktop user management experience.
- Core: add support for WebSockets as an alternative to server-sent events. This introduces a new way for n8n's backend to push changes to the UI. The default is still server-sent events. If you're experiencing issues with the UI not updating, try changing to WebSockets by setting the `N8N_PUSH_BACKEND` environment variable to `websocket`.
- Editor: add autocomplete for objects.
- Editor: add autocomplete for expressions to the HTML editor component.

## Node enhancements

- Edit Image node: add support for WebP image format.
- HubSpot Trigger node: add conversation events.

## Bug fixes

- Core: disable transactions on SQLite migrations that use PRAGMA `foreign_keys`.
- Core: ensure expression extension doesn't fail with optional chaining.
- Core: fix import command for workflows with old format (affects workflows created before user management was introduced).
- Core: stop copying icons to cache.
- Editor: prevent creation of input connections for nodes without input slot.
- Error workflow now correctly checks for subworkflow permissions.
- ActiveCampaign Node: fix additional fields not being sent when updating account contacts.
- Linear Node: fix issue with Issue States not loading correctly.
- MySQL migration parses database contents if necessary (fix for MariaDB).

## Contributors

[Kirill](#)

## n8n@0.214.3

[View the commits](#) for this version.

**Release date:** 2023-02-09

This is a bug fix release.

## Bug fixes

Editor: prevent creation of input connections for nodes without input slot.

# n8n@0.214.2

[View the commits](#) for this version.

**Release date:** 2023-02-06

This is a bug fix release.

## Bug fixes

- Editor: correctly show OAuth reconnect button.
- Editor: fix resolvable highlighting for HTML editor.

# n8n@0.214.1

[View the commits](#) for this version.

**Release date:** 2023-02-06

This is a bug fix release. It also contains an overhaul of the Slack node.

## Node enhancements

This release includes an overhaul of the Slack node, adding new operations and a better user interface.

## Bug fixes

- Editor: fix an issue with mapping to empty expression input.
- Editor: fix merge node connectors.
- Editor: fix multiple-output endpoints success style after connection is detached.

# n8n@0.214.0

[View the commits](#) for this version.

**Release date:** 2023-02-03

This release contains new features, node enhancements, and bug fixes. The expressions editor now supports autocomplete for some [built in data transformation functions](#). The new features also include two of interest to node builders: a way to allow users to drag and drop data keys, and the new HTML editor component.

## New features

## Autocomplete in the Extension editor

[Data transformation functions](#) now have autocomplete support in the Expression editor.

- Core: export OpenAPI spec for external tools.
- Core: set custom Cache-Control headers for static assets.
- Core: simplify pagination in declarative node design.
- Editor: support mapping keys with drag and drop. Any field with the hint **Enter the field name as text** should now support mapping a data key using drag and drop. Node builders can enable this in their own nodes. Refer to [Creating nodes | UI elements](#) for more information.
- Editor: add the [HTML editor component](#) for use in parameters. This means node builders can now use the HTML editor that n8n uses in the HTML node as a UI component.
- Editor: append expressions in fixed values when mapping to string and JSON inputs.
- Editor: continue to show mapping tooltip after dismiss.
- Editor: roll out schema view.

## Node enhancements

- FTP Node: stream binary data for uploads and downloads.
- Notion Node: add support for image blocks.
- OpenAI Node: add **Frequency Penalty** and **Presence Penalty** to the node options for the text resource.
- Salesforce Node: add **Has Opted Out Of Email** field to lead resource options.
- SSH Node: stream binary data for uploads and downloads.
- Write Binary File Node: stream binary data for writes.
- YouTube Node: switch upload operation over to streaming and resumable uploads API.

## Bug fixes

- Add paired item to the most used nodes.
- Core: fix OAuth2 client credentials not always working.
- Core: fix populating of node custom API call options.
- Core: fix value resolution in declarative node design.
- Core: prevent shared user details being saved alongside execution data.
- Core: revert custom API option injecting.
- Editor: add SMTP info translation link slot.
- Editor: change executions title to match menu.
- Editor: fix JSON field completions while typing.
- Editor: handling router errors when navigation is canceled by user.
- Editor: set max width for executions list.
- Editor: stop unsaved changes popup display when navigating away from an untouched workflow.
- Editor: fix workflow executions view.
- Invoice Ninja Node: fix line items not being correctly set for quotes and invoices.
- Linear Node: fix pagination issue for get all issues.
- Mailchimp Trigger Node: fix webhook recreation.
- Prevent unnecessarily touching updatedAt when n8n starts.
- Schedule Trigger Node: change scheduler behaviour for intervals days and hours.
- Set Node: fix behaviour when selecting continueOnFail and

`pairedItem.`

## n8n@0.213.0

View the [commits](#) for this version.

**Release date:** 2023-01-27

This release introduces LDAP, and a new node for working with HTML in n8n. It also contains node enhancements and bug fixes.

### New features

#### LDAP

This release introduces support for LDAP on Self-hosted Enterprise and Cloud Enterprise plans. Refer to [LDAP](#) for more information on this feature.

- Simplify the Node Details View by moving authentication details to the Credentials modal.
- Improve workflow list performance.

### New nodes

#### HTML node

n8n has a new [HTML node](#). This replaces the HTML Extract node, and adds new functionality to generate HTML templates.

### Node enhancements

- GitLab node: add file resource and operations.
- JIRA Software node: introduce the resource locator component to improve UX.
- Send Email node: this node has been overhauled.

### Bug fixes

- Core: don't crash express app on unhandled rejected promises.
- Core: handle missing binary metadata in download URLs.
- Core: upsert (update and insert) credentials and workflows in the `import:` commands.
- Core: validate numeric IDs in the public API.
- Editor: don't request workflow data twice when opening a workflow.
- Editor: execution list micro optimization.
- Editor: fix node authentication options ordering and hiding options based on node version.
- Editor: fix save modal appearing after duplicating a workflow.
- Editor: prevent workflow execution list infinite no network error.
- Extension being too eager and making calls when it shouldn't.
- Google Drive Node: use the correct MIME type on converted downloads.
- HelpScout Node: fix tag search not working when getting all conversations.
- Notion (Beta) Node: fix create database page with multiple relation

IDs not working.

- Update **Sign in with Google** button to properly match design guidelines.

## Contributors

- [Devin Buhl](#)
- [Sven Ziegler](#)

## n8n@0.212.1

View the [commits](#) for this version.

**Release date:** 2023-01-23

This release includes an overhaul of the Google Analytics node, and bug fixes.

### Node enhancements

This release includes an overhaul of the Google Analytics node. This brings the node's code and components in line with n8n's latest node building styles, and adds support for GA4 properties.

### Bug fixes

- Add schema to Postgres migrations.
- Core: fix execute-once incoming data handling.
- Core: fix expression extension miss-detection.
- Core: fix onWorkflowPostExecute not being called.
- Core: fix URL in error handling for the error Trigger.
- Core: make pinned data with webhook responding on last node manual-only.
- Editor: making parameter input components label configurable.
- Editor: remove infinite loading in not found workflow level execution.
- Linear Node: fix issue with single item not being returned.
- Notion (Beta) Node: fix create database page fails if relation parameter is empty/undefined.

## n8n@0.212.0

View the [commits](#) for this version.

**Release date:** 2023-01-19

This release contains enhancements to the Item Lists node, and bug fixes.

### New features

This release adds experimental support for more Prometheus metrics. Self-hosting users can configure Prometheus using [environment variables](#).

### Node enhancements

The Item Lists node now supports a Summarize operation. This acts similarly to generating pivot tables in Excel, allowing you to aggregate and compare data.

## Bug fixes

- Core: revert a lint rule @typescript-eslint/prefer-nullish-coalescing.
- Editor: allow special characters in node selector completion.
- GitLab Node: update the credential test endpoint.
- Gmail Trigger Node: resolve an issue that was preventing filter by labels from working.
- HTTP Request Node: ensure node enforces the requirement for valid JSON input.
- HTTP Request Node: convert responses to text for all formats, including JSON.

## Contributors

[Sven Ziegler](#)

## n8n@0.211.2

View the [commits](#) for this version.

**Release date:** 2023-01-17

This release contains a bug fix for community nodes, and a new trigger node.

## New nodes

### Google Sheets Trigger node

This release adds a new [Google Sheets Trigger node](#). You can now start workflows in response to row changes or new rows in a Google Sheet.

## Bug fixes

Fixes an issue that was preventing users from installing community nodes.

## n8n@0.211.1

View the [commits](#) for this version.

**Release date:** 2023-01-16

This is a bug fix release. It resolves major issues with 0.211.0.

## New features

Editor: suppress validation errors for freshly added nodes.

## Node enhancements

- Google Ads node: update the API version to 11.
- Google Drive Trigger node: start using the resource locator component.

## Bug fixes

- Build CLI to fix Postgres and MySQL test runs.
- Extend date functions clobbering plus/minus.
- Extension deep compare'n't quite working for some primitives.
- Upgrade jsonwebtoken to address CVE-2022-23540.

# n8n@0.211.0

View the [commits](#) for this version.

**Release date:** 2023-01-13

## New features

- Add demo experiment to help users activate.
- Editor: Improvements to the **Executions** page.
- Editor: Remove prevent-ndv-auto-open feature flag.
- Editor: Update callout component design.
- Add the expression extension framework.

## Bug fixes

- Core: Fixes event message confirmations if no subscribers present.
- Core: Remove threads package, rewrite log writer worker.
- Core: Throw error in UI on expression referencing missing node but don't fail execution.
- DB revert command shouldn't run full migrations before each revert.
- Editor: Disable data pinning on multiple output node types.
- Editor: Don't overwrite window.onerror in production.
- Editor: Execution page bug fixes.
- Editor: Fixes event bus test.
- Editor: Hide data pinning discoverability tooltip in execution view.
- Editor: Mapping tooltip dismiss.
- Editor: Recover from unsaved finished execution.
- Editor: Setting NDV session ID.
- First/last being extended on proxy objects.
- Handle memory issues gracefully.
- PayPal Trigger Node: Omit verification in sandbox environment.
- Report app startup and database migration errors to Sentry.
- Run every database migration inside a transaction.
- Upgrade class-validator to address CVE-2019-18413.
- Zoom Node: Add notice about deprecation of Zoom JWT app support.

## Known issues

You may encounter errors when using the [optional chaining operator](#) in expressions. If this happens, avoid using the operator for now.

# n8n@0.210.2

[View the commits](#) for this version.

**Release date:** 2023-01-09

## New features

### Typeahead for expressions

When using [expressions](#), n8n will now offer you suggestions as you type.

[Image: “Animated GIF showing typeahead in action”]

## Bug fixes

- Core: fix crash of manual workflow executions for unsaved workflows.
- Editor: omit pairedItem from proxy completions.
- Editor: prevent refresh on submit in credential edit modal.
- Google Sheets Node: fix for auto-range detection.
- Read Binary File Node: don't crash the execution when the source file doesn't exist.
- Remove anonymous ID from tracking calls.
- Stop OOM crashes in Execution Data pruning.
- Update links for user management and SMTP help.

## n8n@0.210.1

[View the commits](#) for this version.

**Release date:** 2023-01-05

This is a bug fix release. It also contains a new feature to support user management without SMTP set up.

## New features

### Invite link for users on self-hosted n8n

In earlier versions of self-hosted n8n, you needed SMTP set up on your n8n instance for user management to work. User management required SMTP to sent invitation emails.

0.210.1 introduces an invite link, which you can copy and send to users manually. n8n still recommends setting up SMTP, as this is needed for password resets.

## Bug fixes

- Google Sheets node: fix an issue that was causing append and update operations to fail for numeric values.
- Resolve issues with external hooks.

## n8n@0.210.0

[View the commits](#) for this version.

**Release date:** 2023-01-05

This release introduces two major new features: log streaming and security audits. It also contains node enhancements, bug fixes, and performance improvements.

## New features

### Log streaming

This release introduces [log streaming](#) for users on Enterprise self-hosted plans and custom Cloud plans. Log streaming allows you to send events from n8n to your own logging tools. This allows you to manage your n8n monitoring in your own alerting and logging processes.

### Security audit

This release adds a [security audit](#) feature. You can now run a security audit on your n8n instance, to detect common security issues.

- Core: add support for Redis 6+ ACLs system using username in queue mode. Add the `QUEUE_BULL_REDIS_USERNAME` environment variable.

## Node enhancements

- Compare Datasets node: add an option for fuzzy compare.

## Bug fixes

- Apply credential overwrites recursively. This ensures that overwrites defined for a parent credential type also apply to all credentials extending it.
- Core: enable full manual execution of a workflow using the error trigger.
- Core: fix OAuth credential creation using the API.
- Core: fix an issue with workflow lastUpdated field.
- Editor: clear node creator and scrim on workspace reset.
- Editor: fix an infinite loop while loading executions that aren't on the current executions list.
- Editor: make node title non-editable in executions view.
- Editor: prevent scrim on executable triggers.
- Editor: support tabbing away from inline expression editor.
- Fix executions bulk deletion.
- Google Sheets Node: fix exception when no **Values to Send** are set.
- Respond to Webhook Node: fix issue that caused the content-type header to be overwritten.
- Slack Node: add missing channels:read OAuth2 scope.

## Performance improvements

- Lazy-load public API dependencies to reduce baseline memory usage.
- Lazy-load queue mode and analytics dependencies.

## Contributors

[Thomas S.](#)

## n8n@0.209.4

[View the commits](#) for this version.

**Release date:** 2022-12-28

This is primarily a bug fix release.

### Bug fixes

- Editor: add sticky note without manual trigger.
- Editor: display default missing value in table view as undefined.
- Editor: fix displaying of some trigger nodes in the creator panel.
- Editor: fix trigger node type identification on add to canvas.
- Editor: add the usage and plans page to Desktop.

### New features

Editor: pressing = in an empty parameter input switches to expression mode.

## n8n@0.209.3

[View the commits](#) for this version.

**Release date:** 2022-12-27

This is primarily a bug fix release.

### Bug fixes

- Core: don't send credentials to browser console.
- Core: permit a workflow user who isn't the owner to use their own credentials.
- Editor: fix for loading executions that aren't on the current executions list.
- Editor: make the tertiary button on the **Usage** page transparent.
- Editor: update credential owner warning when sharing.

### New features

Editor: Improve UX for brace completion in the inline expressions editor.

### Node enhancements

Webhook node: when test the node by selecting **Listen For Test Event** then dispatching a call to the webhook, n8n now only runs the Webhook node. Previously, n8n ran the entire workflow. You can still test the full workflow by selecting **Execute Workflow**, then dispatching a test call.

## n8n@0.209.2

View the [commits](#) for this version.

**Release date:** 2022-12-23

This is a bug fix release.

## Bug fixes

- Editor: ensure full tree on expression editor parse. This resolves an issue with the expressions editor cutting off results.
- Fix automatic credential selection when credentials are shared.

## Performance improvements

Improvements to the workflows list performance.

# n8n@0.209.1

View the [commits](#) for this version.

**Release date:** 2022-12-22

This is a bug fix release.

## Bug fixes

- Editor: fix for executions preview scroll load bug and wrong execution being displayed.
- Editor: force parse on long expressions.
- Editor: restore trigger to the nodes panel.
- Nodes: AWS DynamoDB Node Fix issue pagination and simplify issue.
- Nodes: fix DynamoDB node type issues.
- Resolve an issue with credentials and workflows not being matched correctly due to incorrect typing.
- Restore missing tags when retrieving a workflow.

## Contributors

[Nathan Apter](#)

# n8n@0.209.0

View the [commits](#) for this version.

**Release date:** 2022-12-21

This release introduces workflow sharing, and changes to licensing and payment plans.

## New features

### Workflow sharing

This release introduces workflow sharing for users on some plans. With workflow sharing, users can invite other users on the same n8n instance to use and edit their workflows. Refer to [Workflow sharing](#) for details.

## Bug fixes

- Editor: Correctly display trigger nodes without actions and with related regular node in the “On App Events” category.
- Fix stickies resize.
- Hide trigger tooltip for nodes with static test output.
- Keep expression when dropping mapped value.
- Prevent keyboard shortcuts in expression editor modal.
- Redirect home to workflows always.
- Update mapping GIFs.
- Upgrade amqplib to address CVE-2022-0686.
- View option for binary-data shouldn’t download the file on Chrome/Edge.

## n8n@0.208.1

View the [commits](#) for this version.

**Release date:** 2022-12-19

This is a bug fix release.

## Bug fixes

- Always retain original errors in the error chain on NodeOperationError.
- BinaryDataManager should store metadata when saving from buffer.
- Editor: fix for wrong execution data displayed in executions preview.
- Pick up credential test functions from versioned nodes.

## n8n@0.208.0

View the [commits](#) for this version.

**Release date:** 2022-12-16

This release introduces a new inline expressions editor, and a new node: OpenAI. It also contains updates and bug fixes.

## New features

### Inline expression editor

You can now quickly write expressions inline in a node parameter. You can still choose to open the full expressions editor.

[Image: “Screenshot of the inline expressions editors”]

- Add workflow sharing telemetry.
- Core: allow for hiding page usage with environment variables (for upcoming feature)
- Editor: update UI copy for user management setup when sharing is disabled.
- Editor: hide credentials password values.
- Editor: set **All workflows** view as default view on the **Workflows** page.

- Editor: update UI copy for workflow overwriting message.

## New nodes

### Open AI node

This release adds an integration with OpenAI. Refer to the [OpenAI node documentation](#) for details.

### Node enhancements

Send Email node: add support for a “Reply to” email address.

### Bug fixes

- Core: fix for Google and Microsoft generic OAuth2 credentials.
- Core: fix HTTP Digest Auth for responses without an opaque parameter.
- Disqus node: fix thread parameter for “Get All Threads” operation.
- Don’t crash the server when Telemetry is blocked using DNS.
- Editor: allow mapping onto expression editor with selection range.
- Editor: don’t show actions dialog for actionless triggers when selected using keyboard.
- Editor: fix an issue where some node actions wouldn’t select default parameters correctly.
- Editor: fix typo in retry-button option “Retry with original workflow”.
- Update permission for showing workflow caller policy.
- Update pnpm-lock to fix build.

## Contributors

[Daemonxiao](#)

[Kirill](#)

[Ricardo Duarte](#)

## n8n@0.207.1

View the [commits](#) for this version.

**Release date:** 2022-12-13

This is a bug fix release. It resolves an issue with undo.

## n8n@0.207.0

View the [commits](#) for this version.

**Release date:** 2022-12-12

This release adds support for undo/redo actions on the canvas, and includes bug fixes.

## New features

### Undo/redo

You can now undo and redo actions on the canvas.

Use `ctrl/cmd + z` to undo, `ctrl/cmd + shift + z` to redo.

Currently, n8n supports undo/redo for the following canvas actions:

- Adding nodes
- Deleting nodes
- Adding connections
- Deleting connections
- Moving nodes
- Moving connections
- Import workflow (from file/from URL)
- Copy/paste nodes
- Renaming nodes
- Duplicating nodes
- Disabling/enabling nodes
- App integration actions are now displayed in the nodes pane.
- Add sharing permissions info for workflow sharees.
- Handle sharing features when the user skips instance owner setup.
- Update the credential test error message for credential sharees.

## Bug fixes

- Core: remove nodeGetter.
- Core: Increase workflow reactivation max timeout to one day.
- Core: Resolve an issue listing executions with Postgres.
- Core: Remove foreign credentials when copying nodes or duplicating workflow.
- Core: upgrade sse-channel to mitigate CVE-2019-10744.
- Core: use license-sdk v1.6.1.
- Editor: avoid adding Manual Trigger node when webhook node is added.
- Editor: fix credential sharing issues handler when no matching ID or name.
- Editor: fix for broken tab navigation.
- Editor: schema view shows checkbox in case of empty data.
- Editor: Stop returning UNKNOWN ERROR in the response if an actual error message is available.
- Editor: update duplicate workflow action.
- Move Binary Data Node: stringify objects before encoding them in MoveBinaryData.
- Split In Batches Node: fix issue with pairedItem.

## n8n@0.206.1

View the [commits](#) for this version.

**Release date:** 2022-12-06

This is a bug fix release.

## Bug fixes

- Core: make expression resolution improvements.
- Editor: schema unit test stub for Font Awesome icons.
- Remove unnecessary console message.

## n8n@0.206.0

View the [commits](#) for this version.

**Release date:** 2022-12-06

This release contains bug fixes, node enhancements, and a new node input view: schema view.

### New features

#### Schema view

Schema view is a new node input view. It helps you browse the structure of your data, using the first input item.

- Core: add workflow execution statistics.
- Editor: add the alert design system component.
- Editor: fix checkbox line height and make checkbox label clickable.
- Nodes: add a message for read-only nodes.
- Nodes: add a prompt to overwrite changes when concurrent editing occurs.

### Node enhancements

KoBo Toolbox node: add support for the media file API.

### Bug fixes

- Core: fix linter error.
- Core: fix partial execution with pinned data on child node run.
- Core: OAuth2 scopes now save.
- Enable source-maps on WorkflowRunnerProcess in own mode.
- Handle error when workflow doesn't exist or is inaccessible.
- Make nodes.exclude and nodes.include work with lazy-loaded nodes.
- Code Node: restore pairedItem to required n8n item keys.
- Execute Workflow Node: update Execute Workflow node info notice text.
- Gmail Trigger Node: trigger node missing some emails.
- Local File Trigger Node: fix issue that causes a crash if the ignore field is empty.

### Contributors

[Marcel](#)

[Yann Jouanique](#)

## n8n@0.205.0

View the [commits](#) for this version.

**Release date:** 2022-12-02

This release contains an overhaul of the expressions editor, node enhancements, and bug fixes.

### New features

## **Expressions editor usability overhaul**

This release contains usability enhancements for the expressions editor. The editor now includes color signals to indicate when syntax is valid or invalid, and better error messages and tips.

## **Node enhancements**

- Facebook Graph API node: update to support API version 15.
- Google Calendar node: introduce the resource locator component to help users retrieve calendar parameters.
- Postmark Trigger node: update credentials so they can be used with the HTTP Request node (for custom API calls).
- Todoist node: update to use API version 2.

## **Bug fixes**

- Core: ensure executions list is properly filtered for all users.
- Core: fix `$items().length` in Execute Once mode.
- Core: mark binary data to be deleted when pruning executions.
- Core: OAuth2 scope saved to database fix.
- Editor: fix slots rendering of NodeCreator's NoResults component.
- Editor: JSON view values can be mapped like keys.
- AWS SNS Node: fix a pagination issue.
- Google Sheets Node: fix exception if no matching rows are found.
- Google Sheets Node: fix for append operation if no empty rows in sheet.
- Microsoft Outlook Node: fix binary attachment upload.
- Pipedrive Node: resolve properties not working.
- Lazy load nodes for credentials testing.
- Credential overwrites should take precedence over credential default values.
- Remove background for resource ownership selector.
- Update padding for resource filters dropdown.
- Update size of select components in filters dropdown.
- Update workflow save button type and design and share button type.

## **n8n@0.204.0**

View the [commits](#) for this version.

**Release date:** 2022-11-24

This release contains performance enhancements and bug fixes.

## **New features**

- Core: lazy-load nodes and credentials to reduce baseline memory usage.
- Core: use longer stack traces when error reporting is enabled.
- Dev: add credentials E2E test suite and page object.

## **Bug fixes**

- Core: fix `$items().length` behavior in executeOnce mode.
- Core: fix for unused imports.
- Core: use CredentialsOverwrites when testing credentials.

- Core: disable workflow locking due to issues.
- Editor: fix for missing node connections in dev environment.
- Editor: fix missing resource locator component.
- Editor: prevent node-creator tabs from showing when toggled by CanvasAddButton.
- Editor: table view column limit tooltip.
- Editor: fix broken n8n-info-tip slots.
- IF Node: fix “Is Empty” and “Is Not Empty” operation failures for date objects.
- Remove redundant `await` in nodes API request functions without try/catch.
- Schedule Trigger Node: fixes inconsistent behavior with cron and weekly intervals.
- Workflow activation shouldn’t crash if one of the credential is invalid.

## n8n@0.203.1

View the [commits](#) for this version.

**Release date:** 2022-11-18

This is a bug fix release. It resolves an issue with the Google Sheets node versioning.

## n8n@0.203.0

View the [commits](#) for this version.

**Release date:** 2022-11-17

This release includes an overhaul of the Google Sheets node, as well as other new features, node enhancements, and bug fixes.

### New features

- Add duplicate workflow error handler.
- Add workflow data reset action.
- Add credential runtime checks and prevent tampering during a manual run.

### Node enhancements

- Compare Datasets: UI copy changes to improve usability.
- Google Sheets: n8n has overhauled this node, including improved lookup for document and sheet selection.
- Notion (beta) node: use the resource locator component for database and page parameters.

### Bug fixes

- Core: deduplicate error handling in nodes.
- Editor: show back mapping hint when parameter is focused.
- Editor: add **Stop execution** button to execution preview.
- Editor: curb direct item access linting.
- Editor: fix expression editor variable selector filter.
- Editor: fix for execution retry dropdown not closing.
- Editor: fix for logging error on user logout.

- Editor: fix zero treated as missing value in resource locator.
- Editor: hide pin data in production executions.
- Editor: skip optional chaining operators in Code Node editor linting.
- Editor: update to **Expression/Fixed** toggle - keep expression when switching to **Fixed**.
- Editor: fix foreign credentials being shown for new nodes.
- Editor: store copy of workflow in `workflowsById` to prevent node data bugs.
- Editor: fix user redirect to signin bug.

## n8n@0.202.1

View the [commits](#) for this version.

**Release date:** 2022-11-10

This is a bug fix release. It removes some error tracking.

## n8n@0.202.0

View the [commits](#) for this version.

**Release date:** 2022-11-10

This release contains core product improvements and bug fixes.

### New features

- API: report unhandled app crashes using Sentry.
- API: set up error tracking using Sentry.
- Core: Add ownership, sharing and credential details to GET `/workflows` in n8n's internal API.
- Editor: when building nodes, you can now add a property with type notice to your credentials properties. This was previously available in nodes but not credentials. Refer to [Node UI elements](#) for more information.

### Bug fixes

- API: Don't use names for type ORM connections.
- Core: Fix manual execution of pinned trigger on main mode.
- Core: Streamline multiple pinned triggers behavior.
- Editor: Curb argument linting for `$input.first()` and `$input.last()`
- Editor: Fix duplicate bug when new workflow is open.
- Editor: Fix for incorrect execution saving indicator in executions view.
- Editor: Fix for OAuth authorization.
- Editor: Fix workflow activation from the Workflows view.
- Editor: Fix workflow back button navigation.
- Editor: Prevent adding of the start node when importing workflow in the demo mode.
- Editor: Show string numbers and null properly in JSON view.
- Editor: Switch CodeNodeEditor linter parser to esprima-next.
- Editor: Tweak dragged mapping state.
- Editor: Update workflow buttons spacings.
- Editor: Use base path in workflow preview component URL.
- HTTP Request Node: Show error cause in the output.
- HTTP Request Node: Use the data in **Put Output in Field** field.

- HubSpot Node: Add notice to HubSpot credentials about API Key Sunset.
- Notion Trigger (Beta) Node: Fix Notion trigger polling strategy.
- Raindrop Node: Update access token URL.
- SendInBlue Trigger Node: Fix typo in credential name.
- Update E2E testing ENV variables.

## Contributors

[feelgood-interface](#)  
[Ugo Bataillard](#)

## n8n@0.201.0

View the [commits](#) for this version.

**Release date:** 2022-11-02

This release contains workflow and node enhancements, and bug fixes.

### New features

- Core: reimplement blocking workflow updates on interim changes.
- Editor: block the UI in node details view when the workflow is listening for an event.
- Performance improvements

### Node enhancements

Venafi TLS Protect Cloud node: make issuing template depend on application.

### Bug fixes

- Core: fix wokflow hashing for MySQL.
- Core: make deepCopy backward compatible.
- Editor: ensure displayOptions received the value from the resource locator component.
- Editor: disable the settings link in executions view for unsaved workflows.
- Editor: ensure forms reliably save.
- Editor: fix issues with interim updates in executions view.
- Editor: fix for node creator search.
- Editor: limit columns in table view to prevent the UI becoming unresponsive in the node details view.

## n8n@0.200.1

View the [commits](#) for this version.

**Release date:** 2022-10-28

This is a bug fix release.

### Bug fixes

- API: do not reset the auth cookie on every request to GET /login.
- AWS SNS Trigger node: add missing jsonParse import.
- Core: avoid callstack with circular dependencies.
- Editor: resolve issues with the executions list auto-refresh, and with saving new workflows.
- Editor: redirect the outdated /workflow path.
- Editor: remove a filter that prevented display of running executions.

## n8n@0.200.0

View the [commits](#) for this version.

**Release date:** 2022-10-27

This release contains improvements to the editor, node enhancements and bug fixes.

### New features

- Core, editor: introduce workflow caller policy.
- Core: block workflow update on interim change.
- Editor: add a read-only state for nodes.
- Editor: add execution previews using the new **Executions** tab in the node view.
- Editor: improvements to node panel search.

### Node enhancements

- Airtable Trigger node: add the resource locator component.
- HTTP Request node: add options for raw JSON headers and queries.
- InvoiceNinja node: add support for V5.
- Write Binary File node: add option to append to a file.

### Bug fixes

- API: validate executions and workflow filter parameters.
- Core: amend typing for jsonParse() options.
- Core: fix predefinedCredentialType in node graph item.
- Core: fix canvas node execution skipping parent nodes.
- Core: fix single node execution failing in main mode.
- Core: set JWT authentication token sameSite policy to lax.
- Core: update to imports in helpers.
- Editor: curb item method linting in single-item mode.
- Editor: stop rendering expressions as HTML.
- Email Trigger node: backport V2 mark-seen-after processing to V1.
- Email Trigger node: improve connection handling and credentials.
- HTTP Request node: fix sending previously selected credentials.
- TheHive node: small fixes.

### Contributors

[Bram Kn](#)  
[Nicholas Penree](#)

# n8n@0.199.0

View the [commits](#) for this version.

**Release date:** 2022-10-21

This release includes new nodes, an improved workflow UI, performance improvements, and bug fixes.

## New features

### New workflow experience

This release brings a collection of UI changes, aimed at improving the workflow experience for users. This includes:

- Removing the Start node, and adding help to guide users to find a trigger node.
- Improved node search.
- New nodes: Manual Trigger and Execute Workflow Trigger.
- Core: block workflow updates on interim changes.
- Core: enable sending client credentials in the body of API calls.
- Editor: add automatic credential selection for new nodes.

## New nodes

### Compare node

The Compare Datasets node helps you compare data from two input streams. You can find documentation for the new node [here](#).

### Execute Workflow Trigger node

The Execute Workflow Trigger starts a workflow in response to another workflow. You can find documentation for the new node [here](#).

### Manual Trigger node

The Manual Trigger allows you to start a workflow by clicking **Execute Workflow**, without any option to run it automatically. You can find documentation for the new node [here](#).

### Schedule Trigger node

This release introduces the Schedule Trigger node, replacing the Cron node. You can find documentation for the new node [here](#).

## Node enhancements

- HubSpot node: you can now use your HubSpot credentials in the HTTP Request node to make a [custom API call](#).
- Rundeck node: you can now use your Rundeck credentials in the HTTP Request node to make a [custom API call](#).

## Bug fixes

- Editor: fix a hover bug in the bottom menu.
- Editor: resolve performance issues when opening a node, or editing a code node, with a large amount of data.
- Editor: ensure workflows always stop when clicking the stop button.
- Editor: fix a bug that was causing text highlighting when mapping data in Firefox.
- Editor: ensure correct linting in the Code node editor.
- Editor: handle null values in table view.
- Elasticsearch node: fix a pagination issue.
- Google Drive node: fix typo.
- HTTP Request node: avoid errors when a response doesn't provide a content type.
- n8n node: fix a bug that was preventing the resource locator component from returning all items.

## Contributors

[AndLLA](#)  
[Nicholas Penree](#)  
[vcrwr](#)

## n8n@0.198.2

View the [commits](#) for this version.

**Release date:** 2022-10-14

This release fixes a bug affecting scrolling through parameter lists.

## n8n@0.198.1

View the [commits](#) for this version.

**Release date:** 2022-10-14

This is a bug fix release.

### Bug fixes

- Editor: change the initial position of the Start node.
- Editor: align JSON view properties with their values.
- Editor: fix BASE\_PATH for Vite dev mode.
- Editor: fix data pinning success source.

### Contributor

[Bram Kn](#)

## n8n@0.198.0

View the [commits](#) for this version.

**Release date:** 2022-10-14

### New features

- Editor: update the expressions display.
- Editor: update the n8n-menu component.

## New nodes

### Code node

This release introduces the Code node. This node replaces both the Function and Function Item nodes. Refer to the [Code node](#) documentation for more information.

### Venafi TLS Protect Cloud Trigger node

Start a workflow in response to events in your Venafi Cloud service.

## Node enhancements

- Citrix ADC node: add Certificate Install operation.
- Kafka node: add a **Use key** option for messages.
- MySQL node: use the resource locator component for table parameters, making it easier for users to browse and select their database fields from within n8n.

## Bug fixes

- Core, Editor: prevent overlap between running and pinning data.
- Core: expression evaluation of processes now respects `N8N_BLOCK_ENV_ACCESS_IN_NODE`.
- Editor: ensure the Axios base URL still works when hosted in a subfolder.
- Editor: fixes for horizontal scrollbar rendering.
- Editor: ensure the menu closes promptly when loading a credentials page.
- Editor: menu UI fixes.
- Box node: fix an issue that was causing the Create Folder operation to show extra items.
- GSuite Admin node: resolve issue that was causing the User Update operation to fail.
- GitLab Trigger node: ensure this node activates reliably.
- HTTP Request node: ensure OAuth credentials work properly with predefined credentials.
- KoboToolbox node: fix the hook logs.
- SeaTable node: ensure link items show in response.
- Zoom node: resolve an issue that was causing missing output items.

## Contributors

[Jakob Backlund](#)  
[Yan Jouanique](#)

## n8n@0.197.1

View the [commits](#) for this version.

**Release date:** 2022-10-10

This is a bug fix release. It resolves an issue with display width on the resource locator UI component.

## n8n@0.197.0

View the [commits](#) for this version.

**Release date:** 2022-10-10

This release includes six new nodes, focused around infrastructure management. It also adds support for drag and drop data mapping in the JSON input view, and includes bug fixes.

### New features

- Core: improve light versioning support in declarative node design.
- Editor UI: data mapping for JSON view. You can now map data using drag and drop from JSON view, as well as table view.

### New nodes

#### AWS Certificate Manager

A new integration with AWS Certificate Manager. You can find the documentation [here](#).

#### AWS Elastic Load Balancing

Manage your AWS load balancers from your workflow using the new AWS Elastic Load Balancing node. You can find the documentation [here](#).

#### Citrix ADC

Citrix ADC is an application delivery and load balancing solution for monolithic and microservices-based applications. You can find the documentation [here](#).

#### Cloudflare

Cloudflare provides a range of services to manage and protect your websites. This new node allows you to manage zone certificates in Cloudflare from your workflows. You can find the documentation [here](#).

#### Venafi nodes

This release includes two new Venafi nodes, to integrate with their Protect TLS service.

### Node enhancements

Crypto node: add SHA3 support.

### Bug fixes

- CLI: cache generated assets in a user-writeable directory.

- Core: prevent excess runs when data is pinned in a trigger node.
- Core: ensure hook URLs always added correctly.
- Editor: a fix for an issue affecting linked items in combination with data pinning.
- Editor: resolve a bug with the binary data view.
- GitHub Trigger node: ensure trigger executes reliably.
- Microsoft Excel node: fix pagination issue.
- Microsoft ToDo node: fix pagination issue.

## Contributors

[Stratos Theodorou](#)

## n8n@0.196.0

View the [commits](#) for this version.

**Release date:** 2022-09-30

This release includes major new features:

- Better item linking
- New built-in variables and methods
- A redesigned main navigation
- New nodes, as well as an overhaul of the HTTP Request node

It also contains bug fixes and node enhancements.

## New features

### Improved item linking

Introducing improved support for item linking (paired items). Item linking is a key concept in the n8n data flow. Learn more in [Data item linking](#).

### Overhauled built-in variables

n8n's [built-in methods and variables](#) have been overhauled, introducing new variables, and providing greater consistency in behavior and naming.

### Redesigned main navigation

We've redesigned the main navigation (the left hand menu) to create a simpler user experience.

### Other new features

- Improved error text when loading options in a node.
- On reset, share unshared credentials with the instance owner.

## New nodes

### n8n node

The [n8n node](#) allows you to consume the n8n API in your workflows.

## **WhatsApp Business Platform node**

The [WhatsApp Business Platform](#) node allows you to use the WhatsApp Business Platform Cloud API in your workflows.

## **Node enhancements**

- HTTP Request node: a major overhaul. It's now much simpler to build a custom API request. Refer to the [HTTP Request node documentation](#) for more information.
- RabbitMQ Trigger node: now automatically reconnects on disconnect.
- Slack node: add the 'get many' operation for users.

## **Bug fixes**

- Build: add typing for SSE channel.
- Build: fix lint issue.
- CLI: add git to all Docker images
- CLI: disable X-Powered-By: Express header.
- CLI: disable CORS on SSE connections in production.
- Core: remove commented out lines.
- Core: delete unused dependencies.
- Core: fix and harmonize documentation links for nodes.
- Core: remove the -forceExit flag from CLI tests.
- Editor: add missing event handler to accordion component.
- Editor: fix Storybook setup.
- Editor: ensure BASE\_URL replacement works correctly on Windows.
- Editor: fix parameter input field focus.
- Editor: make lodash aliases work on case-sensitive file systems.
- Editor: fix an issue affecting copy-pasting workflows into pinned data in the code editor.
- Editor: ensure the run data pagination selector displays when appropriate.
- Editor: ensure the run selector can open.
- Editor: tidy up leftover i18n references in the node view.
- Editor: correct an i18n string.
- Editor: resolve slow loading times for node types, node creators, and push connections in the settings view.
- Nodes: update descriptions in the Merge node
- Nodes: ensure the card ID property displays for completed checklists in the Trello node.
- Nodes: fix authentication for the new verions of WeKan.
- Nodes: ensure form names list correctly in the Wufoo Trigger node.

## **Contributors**

[Cristobal Schlaubitz Garcia](#)

## **n8n@0.195.5**

View the [commits](#) for this version.

**Release date:** 2022-09-23

This is a bug fix release. It fixes an issue with extracting values in expressions.

## n8n@0.195.4

[View the commits](#) for this version.

**Release date:** 2022-09-22

This release:

- Adds the ability to resize the main node panel.
- Resolves an issue with resource locator in expressions.

## n8n@0.195.3

[View the commits](#) for this version.

**Release date:** 2022-09-22

This is a bug fix release.

- Editor: fix an expressions bug affecting numbers and booleans.
- Added support for setting the TDS version in Microsoft SQL credentials.

## n8n@0.195.2

[View the commits](#) for this version.

**Release date:** 2022-09-22

This is a bug fix release. It resolves an issue with MySQL migrations.

## n8n@0.195.1

[View the commits](#) for this version.

**Release date:** 2022-09-21

This is a bug fix release. It resolves an issue with Postgres migrations.

## n8n@0.195.0

[View the commits](#) for this version.

**Release date:** 2022-09-21

This release introduces user management and credential sharing for n8n's Cloud platform. It also contains other enhancements and bug fixes.

### New features

#### User management and credential sharing for Cloud

This release adds support for n8n's existing [user management](#) functionality to Cloud, and introduces a new feature: [credential sharing](#). Credential sharing is currently only available on Cloud.

Also in this release:

- Added a `resourceLocator` parameter type for nodes, and started upgrading n8n's built-in nodes to use it. This new option helps users who need to specify the ID of a record or item in an external service. For example, when using the Trello node, you can now search for a specific card by ID, URL, or do a free text search for card titles. Node builders can learn more about working with this new UI element in n8n's [UI elements](#) documentation.
- Cache npm dependencies to improve performance on self-hosted n8n

## Bug fixes

- Box node: fix an issue that sometimes prevented response data from being returned.
- CLI: prevent n8n from crashing when it encounters an error in poll method.
- Core: prevent calls to constructor, to forbid arbitrary code execution.
- Editor: fix the output panel for Wait node executions.
- HTTP node: ensure instance doesn't crash when batching enabled.
- Public API: corrections to the OAuth schema.
- Xero node: fix an issue that was causing line amount types to be ignored when creating new invoices.

## Contributors

[Ikko Ashimine](#)

## n8n@0.194.0

View the [commits](#) for this version.

**Release date:** 2022-09-15

This release includes new nodes: a Gmail trigger, Google Cloud Storage, and Adalo. It also contains major overhauls of the Gmail and Merge nodes.

## New features

- CLI: load all nodes and credentials code in isolation.
- Core, Editor UI: introduce support for node deprecation.
- Editor: implement HTML sanitization for Notification and Message components.
- Editor: display the input number on multi-input nodes.

## New nodes

### Adalo

Adalo is a low code app builder. Refer to n8n's [Adalo node documentation](#) for more information.

### Google Cloud Storage

n8n now has a [Google Cloud Storage node](#).

## Gmail Trigger

n8n now has a [Gmail Trigger node](#). This allows you to trigger workflows in response to a Gmail account receiving an email.

## Node enhancements

- Gmail node: this release includes an overhaul of the [Gmail node](#), with updated resources and operations.
- Merge node: a major overhaul. Merge mode's have new names, and have been simplified. Refer to the [Merge node documentation](#) to learn more.
- MongoDB node: updated the Mongo driver to 4.9.1.

## Bug fixes

- CLI: core: address Dependabot warnings.
- CLI: avoid scanning unnecessary directories on Windows.
- CLI: load nodes and directories on Windows using the correct file path.
- CLI: ensure password reset triggers internal and external hooks.
- CLI: use absolute paths for loading custom nodes and credentials.
- Core: returnJsonArray helper no longer breaks nodes that return no data.
- Core: fix an issue with node renaming and expressions.
- Core: update OAuth endpoints to use the instance base URL.
- Nodes: resolved an issue that was preventing versioned nodes from loading.
- Public API: better error handling for bad requests.
- AWS nodes: fixed an issue with credentials testing.
- GoogleBigQuery node: fix for empty responses when creating records.
- HubSpot node: correct the node name on the canvas.

## Contributors

[Rhys Williams](#)

## n8n@0.193.5

View the [commits](#) for this version.

**Release date:** 2022-09-07

This is a bug fix release.

## Bug fixes

- Editor: prevent editing in the Function nodes in executions view.
- Editor: ensure button widths are correct.
- Editor: fix a popup title.
- Gmail node: fix an issue introduced due to incorrect automatic data formatting.

## n8n@0.193.4

View the [commits](#) for this version.

**Release date:** 2022-09-06

This release contains new features that lay the groundwork for upcoming releases, and bug fixes.

### New features

- It's now possible to configure the stop time for workers.
- CLI: Added external hooks for when members are added or deleted.
- Editor: Use the i18n component for localization (replacing v-html)

### Bug fixes

- CLI: include "auth-excluded" endpoints on the history middleware as well.
- Core: fix MySQL migration issue with table prefix.
- Correct spelling.
- Fix n8n-square-button import.
- AWS nodes: handle query string and body properly for AWS related requests.
- AWS Lambda node: fix JSON data being sent to AWS Lambda as string.
- Beeminder node: fix request ID not being sent when creating a new data point.
- GitHub node: fix binary data not being returned.
- GraphQL node: fix issue with return items.
- Postgres node: fix issue with Postgres insert and paired item.
- Kafka Trigger node: fix Kafka trigger not working with default max requests value.
- MonicaCrm node: fix pagination when using return all.
- Gmail node: fix bug related to paired items.
- Raindrop node: fix issue refreshing OAuth2 credentials.
- Shopify node: fix pagination when empty fields are sent.

### Contributors

[Aaron Delasy](#)  
[ruanjiefeng](#)

## n8n@0.193.3

View the [commits](#) for this version.

**Release date:** 2022-09-01

This release contains bug fixes and node enhancements.

### Node enhancements

MongoDB node: add credential testing and two new operations.

### Bug fixes

- CLI: only initialize the mailer if the connection can be verified.
- Core: fix an issue with disabled parent outputs in partial executions.
- Nodes: remove duplicate wrapping of paired item data.

## n8n@0.193.2

View the [commits](#) for this version.

**Release date:** 2022-09-01

This is a bug fix release. It resolves an issue that was causing errors with OAuth2 credentials.

## n8n@0.193.1

View the [commits](#) for this version.

**Release date:** 2022-08-31

This is a bug fix release. It resolves an issue that was preventing column headings from displaying correctly in the editor.

## n8n@0.193.0

View the [commits](#) for this version.

**Release date:** 2022-08-31

This release contains a new node, feature enhancements, and bug fixes.

### New nodes

This release adds an integration for [HighLevel](#), an all-in-one sales and marketing platform.

### Enhancements

- Docker: reduce the size of Alpine Docker images.
- Editor: improve mapping tooltip behavior.

### Bug fixes

- Core: make digest auth work with query parameters.
- Editor: send data as query on DELETE requests.
- Fix credentials\_entity table migration for MySQL.
- Improve .npmignore to reduce the size of the published packages.

### Contributors

[pemontto](#)  
[Tzachi Shirazi](#)

## n8n@0.192.2

View the [commits](#) for this version.

**Release date:** 2022-08-25

This is a bug fix release.

## Bug fixes

- Editor: fix the feature flag check when PostHog is unavailable.
- Editor: fix for a mapping bug that occurred when value is null.

# n8n@0.192.1

View the [commits](#) for this version.

**Release date:** 2022-08-25

This is a bug fix release.

## Bug fixes

Account for non-array types in pinData migration.

# n8n@0.192.0

View the [commits](#) for this version.

**Release date:** 2022-08-24

This release contains new features and enhancements, as well as bug fixes.

## New features

### Map nested fields

n8n@0.187.0 saw the first release of [data mapping](#), allowing you to drag and drop top level data from a node's **INPUT** panel into parameter fields. With this release, you can now drag and drop data from any level.

- Core and editor: support pairedItem for pinned data.
- Core and editor: integrate PostHog.
- Core: add a command to scripts making it easier to launch n8n with tunnel.
- CLI: notify external hooks about user profile and password changes.

## Bug fixes

- Core: account for the enabled state in the first pinned trigger in a workflow.
- Core: fix pinned trigger execution.
- CLI: handle unparseable strings during JSON key migration.
- CLI: fix the excessive instantiation type error for flattened executions.
- CLI: initiate the nodes directory to ensure `npm install` succeeds.
- CLI: ensure tsc build errors also cause Turborepo builds to fail.

- Nextcloud node: fix an issue with credential verification.
- Freshdesk node: fix an issue where the getAll operation required non-existent options.

## n8n@0.191.1

View the [commits](#) for this version.

**Release date:** 2022-08-19

This is a bug fix release. It resolves an issue that was causing node connectors to disappear after a user renamed them.

## n8n@0.191.0

View the [commits](#) for this version.

**Release date:** 2022-08-17

This release lays the groundwork for wider community nodes support. It also includes some bug fixes.

### New features

- Community nodes are now enabled based on npm availability on the host system. This allows n8n to introduce community nodes to the Desktop edition in a future release.
- Improved in-app guidance on mapping data.

### Bug fixes

- CLI: fix the community node tests on Postgres and MySQL.
- Core: fix an issue preventing child workflow executions from displaying.
- Editor: handle errors when opening settings and executions.
- Editor: improve expression and parameters performance.
- Public API: fix executions pagination for n8n instances using Postgres and MySQL.

## n8n@0.190.0

View the [commits](#) for this version.

**Release date:** 2022-08-10

This is a bug fix release.

### Bug fixes

- Core: fix a crash caused by parallel calls to test webhooks.
- Core: fix an issue preventing static data being saved for poll triggers.
- Public API: fix a pagination issue.
- GitHub Trigger: typo fix.

### Contributors

[Nathan Poirier](#)

## n8n@0.189.1

View the [commits](#) for this version.

**Release date:** 2022-08-05

This is a bug fix release.

### Bug fixes

Fixed an issue with MySQL and MariaDB migrations.

## n8n@0.189.0

View the [commits](#) for this version.

**Release date:** 2022-08-03

This release includes a new node, Sendinblue, as well as bug fixes.

### New nodes

[Sendinblue node](#) and [Sendinblue Trigger node](#): introducing n8n's [Sendinblue](#) integration.

### Node enhancements

[NocoDB node](#): add support for v0.90.0+

### Bug fixes

- Editor: fix a label cut off.
- Fix an issue with saving workflows when tags are disabled.
- Ensure support for community nodes on Windows.

### Contributors

[mertmit](#)  
[Nicholas Penree](#)

## n8n@0.188.0

View the [commits](#) for this version.

**Release date:** 2022-07-27

This release contains a new node for Metabase, bug fixes, and node and product enhancements.

### New nodes

#### Metabase

This release includes a new [Metabase node](#). Metabase is a business data analysis tool.

## Enhancements

This release includes improvements to n8n's core pairedItems functionality.

## Node enhancements

- Item Lists node: add an operation to create arrays from input items.
- [Kafka Trigger node](#): add more option fields.

## Bug fixes

- Core: add Windows support to `import:credentials --separate`.
- Editor: correct linking buttons color.
- Editor: ensure data pinning works as expected when `pinData` is null.
- Editor: fix a bug with spaces.
- Editor: resolve an issue with sticky note duplication and positioning.
- Editor: restore missing header colors.
- AWS DynamoDB node: fix for errors with expression attribute names.
- Mautic node: fix an authentication issue.
- Rocketchat node: fix an authentication issue.

## Contributors

[Nicholas Penree](#)

## n8n@0.187.2

View the [commits](#) for this version.

**Release date:** 2022-07-21

This is a bug fix release.

- Editor: fix for a console issue.
- Editor: fix a login issue for non-admin users.
- Editor: fix problems with the credentials modal that occurred when no node is open.
- NocoDB node: fix for an authentication issue.

## n8n@0.187.1

View the [commits](#) for this version.

**Release date:** 2022-07-20

This release fixes a bug that was preventing new nodes from reliably displaying in all browsers.

# n8n@0.187.0

View the [commits](#) for this version.

**Release date:** 2022-07-20

This release includes several major new features, including:

- The community nodes repository: a new way to build and share nodes.
- Data pinning and data mapping: accelerate workflow development with better data manipulation functionality.

## New features

### Community nodes repository

This release introduces the [community node repository](#). This allows developers to build and share nodes as npm packages. Users can install community-built nodes directly in n8n.

### Data pinning

[Data pinning](#) allows you to freeze and edit data during workflow development. Data pinning means saving the output data of a node, and using the saved data instead of fetching fresh data in future workflow executions. This avoids repeated API calls when developing a workflow, reducing calls to external systems, and speeding up workflow development.

### Data mapping

This release introduces a drag and drop interface for [data mapping](#), as a quick way to map data without using expressions.

### Simplify authentication setup for node creators

This release introduces a simpler way of handling authorization when building a node. All credentials should now contain an `authenticate` property that dictates how the credential is used in a request. n8n has also simplified authentication types: instead of specifying an authentication type and using the correct interface, you can now set the type as "generic", and use the `IAuthenticateGeneric` interface.

You can use this approach for any authentication method where data is sent in the header, body, or query string. This includes methods like bearer and basic auth. You can't use this approach for more complex authentication types that require multiple calls, or for methods that don't pass authentication data. This includes OAuth.

For an example of the new authentication syntax, refer to n8n's [Asana node](#).

```
// in AsanaApi.credentials.ts
import {
 IAuthenticateGeneric,
 ICredentialType,
 INodeProperties,
} from 'n8n-workflow';
```

```

export class AsanaApi implements ICredentialType {
 name = 'asanaApi';
 displayName = 'Asana API';
 documentationUrl = 'asana';
 properties: INodeProperties[] = [
 {
 displayName: 'Access Token',
 name: 'accessToken',
 type: 'string',
 default: '',
 },
];
}

authenticate: IAuthenticateGeneric = {
 type: 'generic',
 properties: {
 headers: {
 Authorization: '=Bearer
{{$credentials.accessToken}}',
 },
 },
};
}

```

## Other new features

- Added a `preAuthentication` method to credentials.
- Added more credentials tests.
- Introduce automatic fixing for paired item information in some scenarios.

## Node enhancements

- [ERPNext node](#): add credential tests, and add support for unauthorized certs.
- [Google Drive node](#): add support for move to trash.
- [Mindee node](#): support new version.
- [Notion node](#): support ignoring the Notion URL property if empty.
- [Shopify node](#): add OAuth support.

## Bug fixes

- API: add missing node settings parameters.
- API: validate static data value for resource workflow.
- Baserow Node: fix an issue preventing table names from loading.
- Editor: hide the **Execute previous node** button when in read-only mode.
- Editor: hide tabs if there's only one branch.
- Roundup of link fixes in nodes.

## Contributors

[Florian Bachmann](#) [Olivier Aygaleng](#)

**n8n@0.186.1**

View the [commits](#) for this version.

**Release date:** 2022-07-14

This is a bug fix release. It includes a fix for an issue with the Airtable node.

## n8n@0.186.0

View the [commits](#) for this version.

**Release date:** 2022-07-13

This release contains bug fixes and node enhancements.

### New features

- Add item information to more node errors.
- Update multiple credentials with tests, and add support for custom operations.

### Node enhancements

- [AWS DynamoDB node](#): improve error handling and add an optional GetAll Scan FilterExpression.
- [Customer.io node](#): add support for tracking region selection.
- [Elasticsearch node](#): add 'Source Excludes' and 'Source Includes' options to the Document: getAll operation. Add credential tests, index pipelines, and index refresh.
- [Freshworks CRM node](#): add search and lookup functionality.
- [JIRA node](#): add optional query authentication.
- [Postgres node](#): improve handling of large numbers.
- [Redis node](#): add push and pop operations.
- [Rename node](#): add regex replace.
- [Spreadsheet file node](#): allow skipping headers when writing spreadsheets.

### Bug fixes

- Editor: Fix an error that occurred after repeated executions.
- [EmailReadImap node](#): improve handling of network problems.
- [Google Drive node](#): process input items using the list operation.
- [Telegram node](#): fix for a bug affecting sending binary data (images, documents and so on).

### Contributors

[Bryce Sheehan](#) [h4ux](#) [miguel-mconf](#) [Nicholas Penree](#) [pemontto](#) [Yann Jouanique](#)

## n8n@0.185.0

View the [commits](#) for this version.

**Release date:** 2022-07-05

This release adds a new node, Google Ads. It also contains bug fixes and node enhancements, as well as a small addition to core.

## New features

Core: add the `action` parameter to `INodePropertyOptions`. This parameter is now available when building nodes.

## New nodes

[Google Ads node](#): n8n now provides a Google Ads node, allowing you to get data from Google Ad campaigns.

## Node enhancements

- [DeepL node](#): Add support for longer text fields, and add credentials tests.
- [Facebook Graph API node](#): Add support for Facebook Graph API 14.
- [JIRA node](#): Add support for the simplified option with rendered fields.
- [Webflow Trigger node](#): Reduce the chance of webhook duplication. Add a credentials test.
- [WordPress node](#): Add a post template option.

## Bug fixes

- [HubSpot node](#): Fix for search endpoints.
- [KoboToolbox node](#): Improve attachment matching logic and GeoJSON Polygon format.
- [Odoo node](#): Prevent possible issues with some custom fields.
- Sticky note node: Fix an issue that was causing the main header to hide.
- [Todoist node](#): Improve multi-item support.

## Contributors

[cgobrech](#) [pemontto](#) [Yann Jouanique](#) [Zapfmeister](#)

## n8n@0.184.0

View the [commits](#) for this version.

**Release date:** 2022-06-29

This release includes:

- New core features
- Enhancements to the Clockify node.
- Bug fixes.

## New features

- You can now access `getBinaryDataBuffer` in the pre-send method.
- n8n now exposes the item index being processed by a node.
- Migrated the expressions templating engine to [n8n's fork of riot-tmpl](#).

## Node enhancements

[Clockify node](#): added three new resources: Client, User, and Workspace. Also added support for custom API calls.

## Bug fixes

- Core: fixed an error with logging circular links in JSON.
- Editor UI: now display the full text of long error messages.
- Editor UI: fix for an issue with credentials rendering when the node has no parameters.
- [Cortex node](#): fix an issue preventing all analyzers being returned.
- [HTTP Request node](#): ensure all OAuth2 credentials work with this node.
- [LinkedIn node](#): fix an issue with image preview.
- [Salesforce node](#): fix an issue that was causing the lead status to not use the new name when name is updated.
- Fixed an issue with required/optional parameters.

## Contributors

[pemontto](#)

## n8n@0.183.0

View the [commits](#) for this version.

**Release date:** 2022-06-21

This release contains node enhancements and bug fixes, as well as an improved trigger nodes panel.

## New features

Enhancements to the **Trigger** inputs panel: When using a trigger node, you will now see an **INPUT** view that gives guidance on how to load data into your trigger.

## Node enhancements

- [HubSpot node](#): you can now assign a stage on ticket update.
- [Todoist node](#): it's now possible to move tasks between sections.
- [Twake node](#): updated icon, credential test added, and added support for custom operations.

## Bug fixes

- Core: don't allow OPTIONS requests from any source.
- Core: GET /workflows/:id now returns tags.
- Core: ensure predefined credentials show up in the HTTP Request node.
- Core: return the correct error message on Axios error.
- Core: updates to the expressions allow-list and deny-list.

## Contributors

[Bryce Sheehan Rahimli Rahim](#)

## n8n@0.182.1

View the [commits](#) for this version.

**Release date:** 2022-06-16

This is a bug fix release. It resolves an issue with restarting waiting executions.

## n8n@0.182.0

View the [commits](#) for this version.

**Release date:** 2022-06-14

This release contains enhancements to the Twilio and Wise integrations, and adds support for a new grant type for OAuth2. It also includes some bug fixes.

### New features

Added support for the client\_credentials grant type for OAuth2.

### Node enhancements

- [Twilio node](#): added the ability to make a voice call using TTS.
- [Wise node](#): added support for downloading statements as JSON, CSV, or PDF.

### Bug fixes

- Core: fixes an issue that was causing parameters to get lost in some edge cases.
- Core: fixes an issue with combined expressions not resolving if one expression was invalid.
- Core: fixed an issue that was causing the public API to fail to build on Windows.
- Editor: ensure errors display correctly.
- [HTTP Request node](#): better handling for requests that return null.
- [Pipedrive node](#): fixes a limits issue with the GetAll operation on the Lead resource.
- [Postbin node](#): remove a false error.

### Contributors

[Albrecht Schmidt](#) [Erick Friis](#) [JoLo](#) [Shaun](#) [Valentin Mocanu](#)

## n8n@0.181.2

View the [commits](#) for this version.

**Release date:** 2022-06-09

This is a bug fix release. It resolves an issue that was sometimes causing nodes to error when they didn't return data.

## n8n@0.181.1

[View the commits](#) for this version.

**Release date:** 2022-06-09

This is a bug fix release. It fixes two issues with multi-input nodes.

## n8n@0.181.0

[View the commits](#) for this version.

**Release date:** 2022-06-08

This release introduces the public API.

### New feature highlights

#### The n8n public API

This release introduces the n8n public REST API. Using n8n's public API, you can programmatically perform many of the same tasks as you can in the GUI. The API includes a built-in Swagger UI playground. Refer to the [API documentation](#) for more information.

#### Other new features

- **Core:** you can now block user access to environment variables using the `N8N_BLOCK_ENV_ACCESS_IN_NODE` variable.

#### Bug fixes

- **Core:** properly resolve expressions in declarative style nodes.

## n8n@0.180.0

[View the commits](#) for this version.

**Release date:** 2022-06-07

This release adds a new node for Cal.com, support for tags in workflow import and export, UI improvements, node enhancements, and bug fixes.

### New features

#### Tags in workflow import and export

When importing or exporting a workflow, the JSON can now include workflow tags.

#### Improved handling of activation errors

n8n now supports running an error workflow in response to an activation error.

### New nodes

#### Cal.com trigger

This release adds a new trigger node for Cal.com. Refer to the [Cal Trigger documentation](#) for more guidance.

## Node enhancements

- [GitHub node](#): add the Get All operation to the Organization resource.
- [QuickBooks node](#): add a new optional field for tax items.

## Bug fixes

- Restore support for window in expressions.
- Fix to the user-management:reset command.
- Resolve crashes in queue mode.
- Correct delete button hover spacing.
- Resolve a bug causing stuck loading states.
- [EmailReadImap node](#): improve error handling.
- [HubSpot node](#): fix contact loading.

## Contributors

[Mark Steve Samson](#) [Syed Ali Shahbaz](#)

## n8n@0.179.0

View the [commits](#) for this version.

**Release date:** 2022-05-30

This release features a new node for PostBin, as well as various node enhancements and bug fixes.

## New nodes

### PostBin node

PostBin serves as a wrapper for standard HTTP libraries which can be used to test arbitrary API/Webhooks by sending requests and providing more advanced ways to analyze the responses.

## Node enhancements

- [RabbitMQ Trigger node](#): Made message acknowledgement and parallel processing configurable.
- [ServiceNow node](#): Added support for attachments.
- [Todoist node](#): Added support for specifying the parent task when adding and listing tasks.

## Bug fixes

- **Core**: Fixed migrations on non-public Postgres schema.
- **Core**: Mitigated possible XSS vulnerability when importing workflow templates.
- **Editor UI**: fixed erroneous hover state detection close to the sticky note button.
- **Editor UI**: fixed display behavior of credentials assigned to

versioned nodes.

- [Discord node](#): Fixed rate limit handling.
- [Gmail node](#): Fixed sending attachments in filesystem data mode.
- [Google Sheets node](#): Fixed an error preventing the *Use Header Names as JSON Paths* option from working as expected.
- [Nextcloud node](#): Updated the node so the list:folder operation works with Nextcloud version 24.
- [YouTube node](#): Fixed problem with uploading large files.

## n8n@0.178.2

View the [commits](#) for this version.

**Release date:** 2022-05-25

This is a bug fix release. It solves an issue with loading parameters when making custom operations calls.

## n8n@0.178.1

View the [commits](#) for this version.

**Release date:** 2022-05-24

This is a bug fix release. It solves an issue with setting credentials in the HTTP Request node.

## n8n@0.178.0

View the [commits](#) for this version.

**Release date:** 2022-05-24

This release adds support for reusing existing credentials in the HTTP Request node, making it easier to do custom operation with APIs where n8n already has an integration.

The release also includes improvements to the nodes view, giving better detail about incoming data, as well as some bug fixes.

### New features

#### Credential reuse for custom API operations

n8n supplies hundreds of nodes, allowing you to create workflows that link multiple products. However, some nodes don't include all the possible operations supported by a product's API. You can work around this by making a custom API call using the HTTP Request node.

One of the most complex parts of setting up API calls is managing authentication. To simplify this, n8n now provides a way to use existential credential types (credentials associated with n8n nodes) in the HTTP Request node.

For more information, refer to [Custom API operations](#).

#### Node details view

An improved node view, showing more detail about node inputs.

## Node enhancements

[Salesforce Node](#): Add the **Country** field.

## Bug fixes

- [Editor UI](#): don't display the dividing line unless necessary.
- [Editor UI](#): don't display the 'Welcome' sticky in template workflows.
- [Slack Node](#): Fix the kick operation for the channel resource.

# n8n@0.177.0

View the [commits](#) for this version.

**Release date:** 2022-05-17

This release contains node enhancements, an improved welcome experience, and bug fixes.

## New features

### Improved welcome experience

A new [introductory video](#), automatically displayed for new users.

### Automatically convert Luxon dates to strings

n8n now automatically converts Luxon DateTime objects to strings.

## Node enhancements

- [Google Drive Node](#): Drive upload, delete, and share operations now support shared Drives.
- [Microsoft OneDrive](#): Add the rename operation for files and folders.
- [Trello](#): Add support for operations relating to board members.

## Bug fixes

- [core](#): Fix call to /executions-current with unsaved workflow.
- [core](#): Fix issue with fixedCollection having all default values.
- [Edit Image Node](#): Fix font selection.
- [Ghost Node](#): Fix post tags and add credential tests.
- [Google Calendar Node](#): Make it work with public calendars and clean up.
- [KoBoToolbox Node](#): Fix query and sort + use question name in attachments.
- [Mailjet Trigger Node](#): Fix issue that node couldn't get activated.
- [Pipedrive Node](#): Fix resolve properties when using multi option field.

## Contributors

[Cristobal Schlaubitz Garcia](#) [Yann Jouanique](#)

## n8n@0.176.0

View the [commits](#) for this version.

**Release date:** 2022-05-10

This release contains bug fixes and node enhancements.

### Node enhancements

- [Pipedrive node](#): adds support for filters to the Organization: Get All operation.
- [Pushover node](#): adds an HTML formatting option, and a credential test.
- [UProc node](#): adds new tools.

### Bug fixes

- [core](#): a fix for filtering the executions list by waiting status.
- [core](#): improved webhook error messages.
- [Edit Image node](#): node now works correctly with the binary-data-mode ‘filesystem’.

### Contributors

[Albert Kiskorov](#) [Miquel Colomer](#)

## n8n@0.175.1

View the [commits](#) for this version.

**Release date:** 2022-05-03

This is a bug fix release.

### Bug fixes

Fixes a bug in the editor UI related to node versioning.

## n8n@0.175.0

View the [commits](#) for this version.

**Release date:** 2022-05-02

This release adds support for node versioning, along with node enhancements and bug fixes.

### New features

#### Node versioning

0.175.0 adds support for a lightweight method of node versioning. One node can contain multiple versions, allowing small version increments without code duplication. To use this feature, change the version parameter in your node to an array, and add your version numbers, including your existing version. You can then access the version parameter with @version in your displayOptions (to control which version n8n displays). You can also query the version in your execute function using const nodeVersion = this.getNode().typeVersion;.

## Node enhancements

- [Google Sheets node](#): n8n now handles header names formatted as JSON paths.
- [Microsoft Dynamics CRM node](#): add support for regions other than North America.
- [Telegram node](#): add support for querying chat administrators.

## Bug fixes

- **core**: fixed an issue that was causing n8n to apply authentication checks, even when user management was disabled.
- **core**: n8n now skips credentials checks for disabled nodes.
- **editor**: fix a bug affecting touchscreen monitors.
- [HubSpot node](#): fix for search operators.
- [SendGrid node](#): fixed an issue with sending attachments.
- [Wise node](#): respect the time parameter on get: exchangeRate.

## Contributors

[Jack Rudenko](#) [MC Naveen](#) [vcrwr](#)

## n8n@0.174.0

View the [commits](#) for this version.

**Release date:** 2022-04-25

## New features

### Sticky Notes

This release adds Sticky Notes, a new feature that allows you to annotate and comment on your workflows. Refer to the [Sticky Notes](#) for more information.

## Enhancements

- **core**: allow external OAuth connection. This enhancement adds support for connecting OAuth apps without access to n8n.
- All AWS nodes now support AWS temporary credentials.
- [Google Sheets node](#): Added upsert support.
- [Microsoft Teams node](#): adds several enhancements:
  - An option to limit groups to “member of”, rather than retrieving the whole directory.
  - An option to get all tasks from a plan instead of just a group

- member.
- Autocompletion for plans, buckets, labels, and members in update fields for tasks.
- [MongoDB node](#): you can now parse dates using dot notation.

## Bug fixes

- [Calendly Trigger node](#): updated the logo.
- [Microsoft OneDrive node](#): fixed an issue that was preventing upload of files with special characters in the file name.
- [QuickBooks node](#): fixed a pagination issue.

## Contributors

[Basit Ali](#) [Cody Stamps](#) [Luiz Eduardo de Oliveira](#) [Oliver Trajceski](#)  
[pemonto](#) [Ryan Goggin](#)

## n8n@0.173.1

View the [commits](#) for this version.

**Release date:** 2022-04-19

Fixes a bug with the Discord node icon name.

## n8n@0.173.0

View the [commits](#) for this version.

**Release date:** 2022-04-19

## New nodes

### Markdown node

[Markdown node](#): added a new Markdown node to convert between Markdown and HTML.

### Enhancements

**editor**: you can now drag and drop nodes from the nodes panel onto the canvas.

## Node enhancements

- [Discord node](#): additional fields now available when sending a message to Discord.
- [GoogleBigQuery](#): added support for service account authentication.
- [Google Cloud Realtime Database node](#): you can now select a region.
- [PagerDuty node](#): now supports more detail in incidents.
- [Slack node](#): added support for blocks in Slack message update.

## Bug fixes

- **core:** make the email for user management case insensitive.
- **core:** add rawBody for XML requests.
- **editor:** fix a glitch that caused dropdowns to break after adding expressions.
- **editor:** reset text input value when closed with Esc.
- Discourse node: fix an issue that was causing incomplete results when getting posts. Added a credentials test.
- Zendesk Trigger node: remove deprecated targets, replace with webhooks.
- Zoho node: fix pagination issue.

## Contributors

[Florian Metz](#) [Francesco Pongiluppi](#) [Mark Steve Samson](#) [Mike Quinlan](#)

## n8n@0.172.0

View the [commits](#) for this version.

**Release date:** 2022-04-11

## Enhancements

- Changes to the data output display in nodes.

## Node enhancements

Magento 2 Node: Added credential tests. PayPal Node: Added credential tests and updated the API URL.

## Bug fixes

**core:** Luxon now applies the correct timezone. Refer to Luxon for more information.

**core:** fixed an issue with localization that was preventing i18n files from loading.

Action Network Node: Fix a pagination issue and add credentials test.

## Contributors

[Paolo Rechia](#)

## n8n@0.171.1

View the [commits](#) for this version.

**Release date:** 2022-04-06

This is a small bug fix release.

## Bug fixes

- **core:** fix issue with current executions not displaying.
- **core:** fix an issue causing n8n to falsely skip some authentication.
- WooCommerce Node: Fix a pagination issue with the GetAll

operation.

## n8n@0.171.0

View the [commits](#) for this version.

**Release date:** 2022-04-03

This release focuses on bug fixes and node enhancements, with one new feature, and one breaking change to the GraphQL node.

### Breaking change to GraphQL node

The GraphQL node now errors when the response includes an error. If you use this node, you can choose to:

- Do nothing: a GraphQL response containing an error will now cause the workflow to fail.
- Update your GraphQL node settings: set **Continue on Fail** to true to allow the workflow to continue even when the GraphQL response contains an error.

### New features

You can now download binary data from individual nodes in your workflow.

### Enhanced nodes

- [Emelia Node](#): Add Campaign > Duplicate functionality.
- [FTP Node](#): Add option to recursively create directories on rename.
- [Mautic Node](#): Add credential test and allow trailing slash in host.
- [Microsoft Teams Node](#): Add chat message support.
- [Mocean Node](#): Add 'Delivery Report URL' option and credential tests.
- [ServiceNow Node](#): Add basicAuth support and fix getColumns loadOptions.
- [Strava Node](#): Add 'Get Streams' operation.

### Bug fixes

- **core**: Fix crash on webhook when last node did not return data
- [EmailReadImap Node](#): Fix issue that crashed process if node was configured wrong.
- [Google Tasks Node](#): Fix 'Show Completed' option and hide title field where not needed.
- [NocoDB Node](#): Fix pagination.
- [Salesforce Node](#): Fix issue that 'status' did not get used for Case => Create & Update

### Contributors

- [Charles Lecalier](#)
- [d3no](#)
- [Ketan Somvanshi](#)
- [Luis Cipriani](#)
- [pemontto](#)

- [Rhys Williams](#)

## n8n@0.170.0

View the [commits](#) for this version.

**Release date:** 2022-03-27

This release focuses on bug fixes and adding functionality to existing nodes.

### Enhanced nodes

- [Crypto Node](#): Add Generate operation to generate random values.
- [HTTP Request Node](#): Add support for OPTIONS method.
- [Jira Node](#): Add Simplify Output option to Issue > Get.
- [Reddit Node](#): Add possibility to query saved posts.
- [Zendesk Node](#): Add ticket status On-hold.

### Bug fixes

- [core](#): Add logs and error catches for possible failures in queue mode.
- [AWS Lambda Node](#): Fix Invocation Type > Continue Workflow.
- [Supabase Node](#): Send token also using Authorization Bearer; fix Row > Get operation.
- [Xero Node](#): Fix some operations and add support for setting address and phone number.
- [Wise Node](#): Fix issue when executing a transfer.

### Contributors

- [FFTDB](#)
- [Fred](#)
- [Jasper Zonneveld](#)
- [pemontto](#)
- [Sergio](#)
- [TheFSilver](#)
- [Valentin Mocanu](#)
- [Yassine Fathi](#)

## n8n@0.169.0

View the [commits](#) for this version.

**Release date:** 2022-03-20

This release includes:

- New functionality for existing nodes
- A new node for Linear
- Bug fixes
- And a license change!

### New license

This release changes n8n's license, from [Apache 2.0 with Commons Clause](#) to [Sustainable Use License](#).

This change aims to clarify n8n's license terms, and n8n's position as a fair-code project.

Read more about the new license in [License](#).

## New nodes

- [Linear Node](#): Add Linear Node.

## Enhanced nodes

- [HTTP Request Node](#): Allow Delete requests with body.
- [KoBoToolbox Node](#): Add KoBoToolbox Regular and Trigger Node.
- [Mailjet Node](#): Add credential tests and support for sandbox, JSON parameters & variables.
- [Mattermost Node](#): Add support for Channel search.

## Other improvements

- Add support for reading IDs from file with executeBatch command.

## Bug fixes

- [GitHub node](#): Fix credential tests and File List operation.
- [Telegram node](#): Fix sending binary data when disable notification is set.

## Contributors

- [Manuel](#)
- [Marcin Kozey](#)
- [Matthew Walther](#)
- [Yann Jouanique](#)

## n8n@0.168.2

For a comprehensive list of changes, view the [commits](#) for this version.

**Release date:** 2022-03-16

This release contains an important bug fix for 0.168.0. Users on 0.168.0 or 0.168.1 should upgrade to this.

## n8n@0.168.1

For a comprehensive list of changes, view the [commits](#) for this version.

**Release date:** 2022-03-15

A bug fix for user management: fixed an issue with email templates that was preventing owners from inviting members.

## n8n@0.168.0

For a comprehensive list of changes, view the [commits](#) for this version.

**Release date:** 2022-03-14

## New feature: user management

User management in n8n allows you to invite people to work in your self-hosted n8n instance. It includes:

- Login and password management
- Adding and removing users
- Two account types: owner and member

Check out the [user management documentation](#) for more information.

## n8n@0.167.0

For a comprehensive list of changes, view the [commits](#) for this version.

**Release date:** 2022-03-13

## Highlights

### Luxon and JMESPath

0.167.0 adds support for two new libraries:

- [Luxon](#): a JavaScript library for working with date and time
- [JMESPath](#): a query language for JSON

You can use Luxon and JMESPath in the code editor and in expressions.

### New expressions variables

We've added two new variables to simplify working with date and time in expressions:

- \$now: a Luxon object containing the current timestamp. Equivalent to DateTime.now().
- \$today: a Luxon object containing the current timestamp, rounded down to the day. Equivalent to DateTime.now().set({ hour: 0, minute: 0, second: 0, millisecond: 0 }).

### Negative operations in If and Switch nodes

Made it easier to perform negative operations on strings.

This release adds one new operation for numbers:

- Is Not Empty

And the following new operations for strings:

- Not Ends With
- Regex Not Match
- Not Starts With
- Is Not Empty

Additionally, Regex is now labelled Regex Match.

### New node: Redis Trigger

Added a Redis Trigger node, so you can now start workflows based on a Redis event.

- [Redis Trigger](#): Added a Redis Trigger node.

### Core functionality

- Added support for [Luxon](#) and [JMESPath](#).
- Added two new expressions variables, \$now and \$today.
- Added more negative operations for numbers and strings.
- Added a link to the course from the help menu.

### Nodes

- [Facebook Graph API](#): Added support for Facebook Graph API 13.
- [HubSpot](#): Added support for private app token authentication.
- [MongoDB](#): Added the aggregate operation.
- [Redis Trigger](#): Added a Redis Trigger node.
- [Redis](#): Added support for publish operations.
- [Strapi](#): Added support for Strapi 4.
- [WordPress](#): Added status as an option to getAll post requests.

### Bugfixes

- The Google Calendar node now correctly applies timezones when creating, updating, and scheduling all day events.
- Fixed a bug that occasionally caused n8n to crash, or shut down workflows unexpectedly.
- You can now use long credential type names with Postgres.

### Contributors

- [Luiz Eduardo de Oliveira Fonseca](#)
- [Vitaliy Fratkin](#)
- [sol](#)
- [vcrw1](#)
- [FFTDB](#)

## n8n@0.166.0

For a comprehensive list of changes, view the [commits](#) for this version.

**Release date:** 2022-03-08

### New nodes

- [Odoo](#)

### Enhanced nodes

- [Function](#): Added support for items without a JSON key.

## Core functionality

- Added new environment variable `N8N_HIRING_BANNER_ENABLED` to enable/disable the hiring banner.
- Fixed a bug preventing keyboard shortcuts from working as expected.
- Fixed a bug causing tooltips to be hidden behind other elements.
- Fixed a bug causing some credentials to be hidden from the credentials list.

## Bug fixes

- [Baserow](#): Fixed a bug preventing the Sorting option of the Get All operation from working as expected.
- [HTTP Request](#): Fixed a bug causing Digest Authentication to fail in some scenarios.
- [Wise](#): Fixed a bug causing API requests requiring Strong Customer Authentication (SCA) to fail.

## Contributors

[pemonto](#)

## n8n@0.165.0

For a comprehensive list of changes, view the [commits](#) for this version.

**Release date:** 2022-02-28

## New nodes

- [Onfleet](#)

## Enhanced nodes

- [Asana](#): Added Create operation to the Project resource.
- [Mautic](#): Added Edit Contact Points, Edit Do Not Contact List, Send Email operations to Contact resource. Also added new Segment Email resource.
- [Notion \(Beta\)](#): Added support for rollup fields to the Simplify Output option. Also added the Parent ID to the Get All operation of the Block resource.
- [Pipedrive](#): Added Marketing Status field to the Create operation of the Person resource, also added User ID field to the Create and Update operations of the Person resource.

## Core functionality

- Added support for workflow templates.
- Fixed a bug causing credentials tests to fail for versioned nodes.
- Fixed a build problem by adding dependencies `@types/lodash.set` to the `workflow` package and `@types/uuid` to the `core` package.
- Fixed an error causing some resources to ignore a non-standard `N8N_PATH` value.
- Fixed an error preventing the placeholder text from being shown when entering credentials.

- Improved error handling for telemetry-related errors.

## Bug fixes

- Orbit: Fixed a bug causing API requests to use an incorrect workspace identifier.
- TheHive: Fixed a bug causing the Ignore SSL Issues option to be applied incorrectly.

## Contributors

[alexwitkowski](#), [Iñaki Breinbauer](#), [lsemaj](#), [Luiz Eduardo de Oliveira Fonseca](#), [Rodrigo Correia](#), [Santiago Botero Ruiz](#), [Saurabh Kashyap](#), [Ugo Bataillard](#)

## n8n@0.164.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2022-02-20

### Core Functionality

- Fixed a bug preventing webhooks from working as expected in some scenarios.

## n8n@0.164.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2022-02-20

### New nodes

- [Google Chat](#)

### Enhanced nodes

- Grist: Added support for self-hosted Grist instances.
- Telegram Trigger: Added new Extra Large option to Image Size field.
- Webhook: Added new No Response Body option. Also added support for DELETE, PATCH and PUT methods.

### Core Functionality

- Added new database indices to improve the performance when querying past executions.
- Fixed a bug causing the base portion of a URL not to be prepended as expected in some scenarios.
- Fixed a bug causing expressions to resolve incorrectly when referencing non-existent nodes or parameters.

## Contributors

Jhalter5Stones, [Valentina Lilova](#), [thorstenfreitag](#)

## n8n@0.163.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2022-02-13

### Core Functionality

- Fixed a bug preventing OAuth2 authentication from working as expected in some scenarios.

## n8n@0.163.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2022-02-13

### New nodes

- [HaloPSA](#)
- [Linear Trigger](#)
- [Zammad](#)

### Enhanced nodes

- [GitHub](#): Added Reference option to the Get operation of the File resource.
- [Twilio](#): Added Status Callbacks option.
- [uProc](#): Sanitized Data Webhook field description.

### Core Functionality

- Added automatic sorting by relative position to the node list inside the expression editor.
- Added new /workflows/demo page to allow read-only rendering of workflows inside an iframe.
- Added optional /healthz health check endpoint to worker instances.
- Fixed unwanted list autofill behaviour inside the expression editor.
- Improved the GitHub actions used by the nightly Docker image.

### Bug fixes

- [Function](#): Fixed a bug leaving the code editor size unchanged after resizing the window.
- [Function Item](#): Fixed a bug leaving the code editor size unchanged after resizing the window.
- [IF](#): Removed the empty sections left after removing a condition.
- Item Lists: Fixed an erroneous placeholder text.

### Contributors

[Iñaki Breinbauer](#), [Manuel pemontto](#)

## n8n@0.162.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2022-02-06

### Enhanced nodes

- [GitHub](#): Added new List operation to File resource.

### Core Functionality

- Added configurable debug logging for telemetry.
- Added support for defining nodes through JSON. This functionality is in alpha state and breaking changes to the interface can take place in upcoming versions.
- Added telemetry support to page events occurring before telemetry is initialized.
- Fixed a bug preventing errors in sub-workflows from appearing in parent executions.
- Fixed a bug where node versioning would not work as expected.
- Fixed a bug where remote parameters would not load as expected.
- Fixed a bug where unknown node types would not work as expected.
- Prevented the node details view from opening automatically after duplicating a node.
- Removed dependency fibers which is incompatible with the current LTS version 16 of Node.js.

### Bug fixes

- [XML](#): Fixed a bug causing the node to alter incoming data.

### Contributors

[pemontto](#)

## n8n@0.161.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2022-02-01

### Core Functionality

- Added optional debug logging to health check functionality.

## n8n@0.161.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2022-01-30

## Core Functionality

- Added default polling interval for trigger nodes using polling.
- Added support for additional hints below parameter fields.
- Fixed a bug preventing default values from being used when testing credentials.
- Improved the wording in the *Save your Changes?* dialog.

## Bug fixes

- [Airtable](#): Improved field description.
- [Airtable Trigger](#): Improved field description.
- [erpNext](#): Prevented the node from throwing an error when no data is found.
- [Gmail](#): Fixed a bug causing the BCC field to be ignored.
- [Move Binary Data](#): Fixed a bug causing the binary data to JSON conversion to fail when using filesystem-based binary data handling.
- [Slack](#): Fixed a typo in the Type field.

## Contributors

[fabian wohlgemuth](#)

# n8n@0.160.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2022-01-22

## New nodes

- [BambooHR](#)

## Core Functionality

- Fixed a bug preventing the binary data preview from using the full available height and width.
- Fixed a build problem by pinning chokidar version 3.5.2.
- Prevent workflow activation when no trigger is present and introduced a modal explaining production data handling.
- Fixed *Filter by tags* placeholder text used in the Open Workflow modal.

## Bug fixes

- [HTTP Request](#): Fixed a bug causing custom headers from being ignored.
- [Mautic](#): Fixed a bug preventing all items from being returned in some situations.
- [Microsoft OneDrive](#): Fixed a bug preventing more than 200 items

from being returned.

- [Spotify](#): Fixed a bug causing the execution to fail if there are more than 1000 search results, also fixed a bug preventing the Get New Releases operation of the Album resource from working as expected.

## Contributors

[fabian wohlgemuth](#)

## n8n@0.159.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2022-01-18

### Core Functionality

- Temporarily removed debug logging for Axios requests.

## n8n@0.159.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2022-01-16

### New nodes

- [Jenkins](#)

### Enhanced nodes

- [GraphQL](#): Added support for additional authentication methods Basic Auth, Digest Auth, OAuth1, OAuth2, and Query Auth.

### Core Functionality

- Added support for executing workflows without an ID through the CLI.
- Fixed a build problem.
- Fixed a bug preventing the tag description from being shown on the canvas.
- Improved build performance by skipping the node-dev package during build.

### Bug fixes

- [Box](#): Fixed a bug causing some files to be corrupted during download.
- [Philips Hue](#): Fixed a bug preventing the node from connecting to Philips Hue.
- [Salesforce](#): Fixed a bug preventing filters on date and datetime fields from working as expected.
- [Supabase](#): Fixed an erroneous documentation link.

## Contributors

[Phil Clifford](#)

## n8n@0.158.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2022-01-09

### New nodes

- [Microsoft Graph Security](#)
- [SyncroMSP](#)
- [Supabase](#)

### Enhanced nodes

- [Edit Image](#): Added Transparent operation.
- [Kafka](#): Added Use Schema Registry option.
- [Kafka Trigger](#): Added Use Schema Registry option.
- [Redis](#): Added database field to credentials.
- [Salesforce](#): Added Account Number field.

### Core Functionality

- Added new external hook when active workflows finished initializing.
- Fixed a bug preventing the personalisation survey from showing up.
- Improved telemetry.

### Bug fixes

- [Edit Image](#): Fixed a bug causing two items to be returned.
- [iCalendar](#): Fixed a bug preventing dates in January from working as expected.
- [Merge](#): Fixed causing empty binary data to overwrite other binary data on merge.

## Contributors

[Ricardo Georgel](#), [Pierre](#), [Vahid Sebto](#)

## n8n@0.157.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2022-01-03

### Core Functionality

- Fixed a bug where not all nodes could use the new binary data handling.

## n8n@0.157.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2022-01-02

### Enhanced nodes

- [Function](#): The node now prevents unsupported data from being returned.
- [Function Item](#): The node now prevents unsupported data from being returned.
- [HubSpot](#): Added Engagement resource with Create, Delete, Get, and Get All operations.
- [Notion \(Beta\)](#): Upgraded the Notion node: Added Search operation for the Database resource, Get operation for Database Page resource, Archive operation for the Page resource. Also added Simplify Output option and test for credential validity.
- [Wait](#): Added new Ignore Bots option.
- [Webhook](#): Added new Ignore Bots option.

### Core Functionality

- Fixed a bug where a wrong number suffix was used after duplicating nodes.

### Bug fixes

- [HTTP Request](#): Fixed a bug where using Digest Auth would fail.

### Contributors

[pemontto](#)

## n8n@0.156.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-12-25

### Enhanced nodes

- [GitLab Trigger](#): Added new trigger events: Confidential Issue, Confidential Comment, Deployment, Release.
- [Google Drive](#): Added support for downloading and converting native Google files.
- [Kitemaker](#): Added Space ID field to Create operation of Work Item resource.
- [Raindrop](#): Added Parse Metadata option to Create, Update operations of the Bookmark resource.

### Core Functionality

- Added execution ID to workflow.postExecute hook
- Added response body to UI for failed Axios requests

- Added support for automatically removing new lines from Google Service Account credentials
- Added support for disabling the UI using environment variable
- Fixed a bug causing the wrong expression result to be shown for items from an output other than the first
- Improved binary data management
- Introduced Monaco as new UI code editor

## Contributors

[Arpad Gabor](#), [Leo Lou](#), [Manuel](#)

## n8n@0.155.2

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-12-19

## Core Functionality

- Added support for internationalization (i18n). This functionality is currently in alpha status and breaking changes are to be expected.

## n8n@0.154.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-12-19

## Enhanced nodes

- [Plivo](#): Added user agent to all API requests.

## Core Functionality

- Allow deletion of nodes from the canvas using the backspace key
- Fixed an issue causing clicks in the value survey to impact the main view
- Fixed an issue preventing the update panel from closing

## Bug fixes

- [Todoist](#): Fixed a bug where using the additional field Due Date Time on the Task resource would cause the Create operation to fail.

## Contributors

[Mohammed Huzaif](#), [Лебедев Иван](#)

## n8n@0.153.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-12-11

## New nodes

- [Figma Trigger \(Beta\)](#)
- [Workable Trigger](#)

## Enhanced nodes

- [Google Contacts](#): Added Query option to Get All operation, also prevented the node from failing when no contacts are found.
- [HTTP Request](#): Added support for query-based authentication.
- [Home Assistant](#): Added support for loading possible options in the Domain, Service, and Entity ID fields.
- [One Simple API](#): Added support for Social Profile resources.
- [PagerDuty](#): Write scope is now requested upon authentication against the PagerDuty OAuth2 API.

## Core Functionality

- Added frontend for value surveys
- Fixed an issue preventing the recommendation logic from working as expected after selecting a work area
- Fixed an issue where a wrong exit code was sent when running n8n on an unsupported version of Node.js
- Fixed an issue where node options would disappear on hovering when a node isn't selected
- Fixed an issue where the execution id was missing when running n8n in queue mode
- Fixed an issue where execution data was missing when waiting for a webhook in queue mode
- Improved error handling when the n8n port is already in use
- Improved diagnostic events
- Removed toast notification on webhook deletion, added toast notification after node is copied
- Removed default trigger tooltip for polling trigger nodes

## Bug fixes

- [APITemplate.io](#): Fixed a bug where the Create operation on the Image resource would fail when the Download option isn't enabled.
- [HubSpot](#): Fixed authentication for new HubSpot applications by using granular scopes when authenticating against the HubSpot OAuth2 API.
- [HubSpot Trigger](#): Fixed authentication for new HubSpot applications by using granular scopes when authenticating against the HubSpot Developer API.
- [Jira Software](#): Fixed an issue where the Reporter field would not work as expected on Jira Server instances.
- [Salesforce](#): Fixed a typo preventing the value in the amount field of from being saved.

## Contributors

pemontto, Jascha Lülsdorf, Jonathan Bennetts

## n8n@0.152.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-12-04

### New nodes

- [Google Calendar Trigger](#)

### Enhanced nodes

- [Telegram Trigger](#): Added support for downloading images to channel\_post updates.

### Core Functionality

- Added a plus (+) connector to end nodes
- Allowed opening workflows and executions in a new window when using Ctrl + Click
- Enforced type checking for all node parameters
- Fixed a build issue in the custom n8n docker image
- Fixed a memory leak in the UI which could occur when renaming nodes or navigate to another workflow
- Improved stability of internal test workflows
- Improved expression security
- Introduced redirect to a new page and UI error message when trying to open a deleted workflow
- Introduced support for multiple arguments when logging
- Updated the onboarding survey

### Bug fixes

- [Google BigQuery](#): Fixed a bug preventing pagination from working as expected when the Return All option is enabled.
- [RabbitMQ Trigger](#): Added Trigger to the name of the trigger node.
- [Salesforce](#): Fixed a typo affecting the Type field of the Opportunity resource.

### Contributors

Zvonimir Erdelja, m2scared

## n8n@0.151.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-11-26

### New nodes

- [DHL](#)

- [Grafana](#)

## Core Functionality

- Fixed a bug causing connections between nodes to disappear when renaming a newly added node after drawing a connection to its endpoints.
- Fixed a build issue by adding TypeScript definitions for validator.js to CLI package, also fixed a linting issue by removing an unused import.
- Improved the waiting state of trigger nodes to explain when an external event is required.
- Loops are now drawn below their source node.

## Bug fixes

- [Edit Image](#): Fixed an issue preventing the Composite operation from working correctly in some cases.

## Contributors

[Jonathan Bennetts](#)

# n8n@0.150.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-11-19

## Enhanced nodes

- [Jira Software](#): Added Components as an additional field.

## Core Functionality

- Fixed a build issue by pinning rudder-sdk-node version 1.0.6 in CLI package.
- Fixed an issue preventing the `n8n import:workflow --separate` CLI command from finding workflows on Windows.
- Further improved the expression security.
- Moved all nodes into separate directories in preparation for internationalization.
- Removing default headers for PUT and PATCH operations when using Axios.
- Revamped the workflow canvas.

## Bug fixes

- [HTTP Request](#): Fixed an issue causing the wrong Content-Type header to be set when downloading a file.
- [ServiceNow](#): Fixed incorrect mapping of incident urgency and impact values.
- [Start](#): Fixed an issue causing the node to be disabled in a new workflow.
- [Xero](#): Fixed an issue causing the node to only fetch the first page

when querying the Xero API.

## n8n@0.149.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-11-13

### New nodes

- [One Simple API](#)

### Enhanced nodes

- [Edit Image](#): Added Circle Primitive to Draw operation. Also added Composite operation.
- [Zendesk](#): Added check for API credentials validity.
- [Zulip](#): Added additional field Role to the Update operation of the User resource.

### Core Functionality

- Fixed an issue causing an error message to be thrown when executing a workflow through the CLI.
- Improved expression security by limiting the available process properties.
- Improved the behaviour of internal tests executed through the CLI.
- Updated the owner of the node user's home directory in the custom docker image.

### Bug fixes

- [Google Tasks](#): Fixed an issue where the Due Date field had no effect (Update operation) or was unavailable (Create operation).
- [HTTP Request](#): Fixed an issue where the Content-Length header was not calculated and sent when using the a Body Content Type of Form-Data Multipart.
- [Stripe Trigger](#): Fixed an issue preventing the node from being activated when a previously created webhook no longer exists.
- [Toggl Trigger](#): Updated the API URL used by the node.

### Contributors

[GeylaniBerk](#), [Jonathan Bennetts](#)

## n8n@0.148.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-11-05

### New nodes

- [Dropcontact](#)

- [Respond to Webhook](#)

## Enhanced nodes

- [Lemlist](#): Added additional fields to Create operation of Lead resource.
- [Slack](#): Added User Group resource.
- [Todoist](#): Added Update operation to Task resource.
- [Wait](#): Improved descriptions of available Respond options.
- [WooCommerce](#): Added password field to Crate operation of Customer resource.

## Core Functionality

- Added a hook after workflow creation.
- Fixed a build issue with npm v7 by overriding unwanted behaviour through the .npmrc file.
- Fixed an issue preventing unknown node types from being imported.
- Fixed an issue with the UI falsely indicating a credential can't be selected when using SQLite and multiple credentials with the same name exist.

## Bug fixes

- [Stripe](#): Fixed an issue where setting additional Metadata fields would not have the expected effect. Also fixed an issue where pagination would not work as expected.
- [Zendesk](#): Fixed an issue preventing the additional field External ID from being evaluated correctly.

## Contributors

[mizzimizzi](#), [nikozila](#), [Pauline](#)

## n8n@0.147.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-11-03

## Core Functionality

- Fixed a build issue by moving the `chokidar` dependency to a regular dependency.

## n8n@0.147.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-11-03

## New nodes

- [Local File Trigger](#)

## Core Functionality

- Improved the database migration process to reduce memory footprint.
- Fixed an issue with telemetry by adding an anonymous ID.

# n8n@0.146.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-10-29

## New nodes

- [Microsoft Dynamics CRM](#)

## Enhanced nodes

- [Agile CRM](#): Added Filters to Get All operation of Contact and Company resources.
- [Date & Time](#): Ensuring the return values are always of type string.
- [IF](#): Added support for moment types to Date & Time condition.

## Core Functionality

- Added name and ID of a workflow to its settings.
- Added parameter inputs to be multi-line.
- Fixed an issue with declaring proxies when Axios is used.
- Fixed an issue with serializing arrays and special characters.
- Fixed an issue with updating expressions after renaming a node.

## Bug fixes

- [HTTP Request](#): Fixed an issue with the Full Response option not taking effect when used with the Ignore Response Code option.

## Contributors

[Valentina Lilova](#), [Oliver Trajceski](#)

# n8n@0.145.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-10-22

## New nodes

- [AWS Textract](#)
- [Google Drive Trigger](#)

## Enhanced nodes

- [Bitbucket Trigger](#): Added check for credentials validity. Removed deprecated User and Team resources, and added the Workspace resource.
- [GitHub](#): Added check for API credentials validity.
- [Home Assistant](#): Added check for credentials validity.
- [Jira Software](#): Added check for credentials validity.
- [Microsoft OneDrive](#): Added functionality to create folder hierarchy automatically upon subfolder creation.
- [Pipedrive](#): Added All Users option to Get All operation of Activity resource.
- [Slack](#): Increase the Slack default query limit from 5 to 100 in order to reduce number of requests.
- [Twitter](#): Added Tweet Mode additional field to the Search operation of Tweet resource.

## Core Functionality

- Changed `vm2` library version from `3.9.3` to `3.9.5`.
- Fixed an issue with ignoring the response code.
- Fixed an issue with overwriting credentials using environment variables.
- Fixed an issue with using query strings combined with the `x-www-form-urlencoded` content type.
- Introduced telemetry.

## Bug fixes

- [Jira Software](#): Fixed an issue with the Expand option for the Issue resource. Also fixed an issue with using custom fields on Jira Server.
- [Slack](#): Fixed an issue with pagination when loading more than 1,000 channels.
- [Strapi](#): Fixed an issue using the Where option of the Get All operation.
- [WooCommerce](#): Fixed an issue where a wrong postcode field name was used for the Order resource.

## Contributors

[pemontto](#), [rdd2](#), [robertodamiani](#), [Rodrigo Correia](#)

## n8n@0.144.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-10-15

## Enhanced nodes

- [Nextcloud](#): Added Share operation to the File and Folder resources.
- [Zendesk](#): Added support for deleting, listing, getting, and recovering suspended tickets. Added the query option for regular tickets. Added assignee emails, internal notes, and public replies

options to the update ticket operation.

## Core Functionality

- Improved the autofill behaviour on Google Chrome when entering credentials.

## Bug fixes

- [Airtable](#): Fixed an issue with the sort field.
- [Cron](#): Set the version of the cron library to 1.7.2.

## Contributors

[Jonathan Bennetts](#)

## n8n@0.143.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-10-14

## Enhanced nodes

- [Pipedrive](#): Added support for getting activities from deal ID.
- [Facebook Graph API](#): Added support for Facebook Graph API versions 11 and 12.

## Core Functionality

- Fixed a build issue affecting a number of AWS nodes.
- Changed workflows to use credential ids primarily (instead of names), allowing users to have different credentials with the same name.

## Bug fixes

- [FTP](#): Fixed error when opening FTP/SFTP credentials.

## Contributors

[Rodrigo Correia](#)

## n8n@0.142.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-10-07

## New nodes

- [Stop and Error](#)

## Core Functionality

- Fixed overlapping buttons when viewing on mobile.
- Fixed issue with partial workflow executions when Wait node was last.
- Fixed issue with broken non-JSON requests.
- Node errors now only displayed for executing nodes, not disconnected nodes.
- Automatic save when executing new workflows with Webhook node.
- Fixed an issue with how arrays were serialized for certain nodes.
- Fixed an issue where executions could not be cancelled when running in Main mode.
- Duplicated workflows now open in a new window.

## Bug fixes

- [HTTP Request](#): Fixed 'Ignore response code' flag.
- [Rundeck](#): Fixed issue with async loading of credentials.
- [SeaTable](#): Fixed issue when entering a Baser URI with a trailing slash.

## Contributors

[Günther, Tom Klingenberg](#)

## n8n@0.141.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-10-01

## Core Functionality

- Fixed issue with body formatting of x-form-www-urlencoded requests.

## n8n@0.141.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-09-30

## New nodes

- [Grist](#)
- [SeaTable](#)
- [SeaTable Trigger](#)
- [urlscan.io](#)

## Core Functionality

- Performance improvements in Editor UI
- Improved error reporting

## Contributors

[Alex Hall](#), [Tom Klingenberg](#)

## n8n@0.140.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-09-29

### New nodes

- [Splunk](#)

### Enhanced nodes

- [Telegram](#): Added binary data support to the Send Animation, Send Audio, Send Document, Send Photo, Send Video, and Send Sticker operations.

## Core Functionality

- Fixed startup behavior when running n8n in scaled mode (i.e. `skipWebhooksDeregistrationOnShutdown` is enabled).
- Fixed behavior around handling empty response bodies.
- Fixed an issue with handling of refresh tokens.

## Contributors

[pemontto](#)

## n8n@0.139.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-09-23

## Core Functionality

- Bug fixes and improvements for Editor UI.

## n8n@0.139.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-09-22

### New nodes

- [Elastic Security](#)
- [Misp](#)
- [Netlify](#)
- [Netlify Trigger](#)

## **Enhanced nodes**

- [HubSpot Trigger](#): Authentication method changed to OAuth2.
- [Wait](#): Added improved status messages for Wait behavior.

## **Core Functionality**

- Updated node design to include support for versioned nodes.

## **Bug fixes**

- [SendGrid](#): Fixed issue with adding contacts to lists.

## **Contributors**

[Matías Aguirre](#)

# **n8n@0.138.0**

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-09-15

## **New nodes**

- Item Lists
- [Magento 2](#)

## **Enhanced nodes**

- [Baserow](#): Added the following filter options: Contains, Contains Not, Date Before Date, Date After Date, Filename Contains, Is Empty, Is Not Empty, Link Row Has, Link Row Does Not Have, Single Select Equal, and Single Select Not Equal.
- [Pipedrive](#): Added support for Notes on Leads.
- [WeKan](#): Added Sort field to the Card resource.

## **Core Functionality**

- General UX improvements to the Editor UI.
- Fixed an issue with the `PayloadTooLargeError`.

## **Bug fixes**

- [Lemlist](#): Fixed issue where events were not sent in the correct property.
- [Notion](#): Fixed issue listed unnamed databases.

## **Contributors**

[bramknuever](#), [Chris Magnuson](#)

# **n8n@0.137.0**

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-09-05

## New nodes

- [Freshservice](#)

## Enhanced nodes

- [Clockify](#): Added Task resource.
- [HubSpot](#): Added dropdown selection for Properties and Properties with History filters for Get All Deals operations.
- [Mautic](#): Added Campaign Contact resource.
- [MongoDB](#): Added ability to query documents by '\_id'.
- [MQTT](#): Added SSL/TLS support to authentication.
- [MQTT Trigger](#): Added SSL/TLS support to authentication.
- [Salesforce](#): Added File Extension option to the Document resource. Added Type field to Task resource.
- [Sms77](#): Added Voice Call resource. Added the following options to SMS resource: Debug, Delay, Foreign ID, Flash, Label, No Reload, Performance Tracking, TTL.
- [Zendesk](#): Added Organization resource. Added Get Organizations and Get Related Data operations to User resource.

## Core Functionality

- Added execution ID to logs of queue processes.
- Added description to operation errors.
- Added ability for webhook processes to wake waiting executions.

## Bug fixes

- [HubSpot](#): Fixed issue with 'RequestAllItems' API.
- [WordPress](#): Fixed issue with 'RequestAllItems' API only returning the first 10 items.

## Contributors

[André Matthies](#), [DeskYT](#), [Frederic Alix](#), [Jonathan Bennetts](#), [Ketan Somvanshi](#), [Luiz Eduardo de Oliveira Fonseca](#), [TheFSilver](#)

# n8n@0.136.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-08-30

## Enhanced nodes

- [Notion](#): Added handling of Rich Text when simplifying data.

## Core Functionality

- General UI design improvements.

- Improved errors messages during debugging of custom nodes.
- All packages upgraded to TypeScript 4.3.5, improved linting and formatting.

## Bug fixes

- FTP: Fixed issue where incorrect paths were displayed when using the node.
- Wait: Fixed issue when receiving multiple files using On Webhook Call operation.
- Webhook: Fixed issue when receiving multiple files.

## n8n@0.135.3

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-08-27

## Core Functionality

- Fixed Canvas UI inconsistencies when duplicating workflows.
- Added log message during upgrade to indicate database migration has started.
- General improvements to parameter labels and tooltips.

## Contributors

[Kyle Mohr](#)

## n8n@0.135.2

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-08-26

## Core Functionality

- Added expression support for credentials.
- Fixed performance issues when loading credentials.

## n8n@0.135.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-08-23

## Core Functionality

- Fixed an issue where if n8n was shutdown during database migration while upgrading versions, errors would result upon next startup.

## n8n@0.135.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-08-22

### New nodes

- [Form.io Trigger](#)
- [Formstack Trigger](#)
- [Wait](#)

### Core Functionality

- In-node method for accessing binary data is now asynchronous and a helper function for this has been implemented.
- Credentials are now loaded from the database on-demand.
- Webhook UUIDs are automatically updated when duplicating a workflow.
- Fixed an issue when referencing values before loops.

### Bug fixes

- [Interval](#): Fixed issue where entering too large a value (> 2147483647ms) resulted in an interval of 1sec being used rather than an error.

### Contributors

[Aniruddha Adhikary](#), [lublak](#), [parthibanbalaji](#)

## n8n@0.134.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-08-15

### Enhanced nodes

- [AWS DynamoDB](#): Added Scan option to Item > Get All operation.
- [Google Drive](#): Added File Name option to File > Update operation.
- [Mautic](#): Added the following fields to Company resource: Address, Annual Revenue, Company Email, Custom Fields, Description, Fax, Industry, Number of Employees, Phone, Website.
- [Notion](#): Added Timezone option when inserting Date fields.
- [Pipedrive](#): Added the following Filters options to the Deal > Get All operation: Predefined Filter, Stage ID, Status, and User ID.
- [QuickBooks](#): Added the Transaction resource and Get Report operation.

### Core Functionality

- Integrated [Nodelinter](#) in n8n.
- Fix to add a trailing slash (/) to all webhook URLs for proper functionality.

## Bug fixes

- [AWS SES](#): Fixed issue where special characters in the message were not encoded.
- [Baserow](#): Fixed issue where Create operation inserted null values.
- [HubSpot](#): Fixed issue when sending context parameter.

## Contributors

[calvintwr](#), [CFarcy](#), [Jeremie Dokime](#), [Michael Hirschler](#), [Rodrigo Correia](#), [sol](#)

## n8n@0.133.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-08-08

## New nodes

- [Monica CRM](#)

## Enhanced nodes

- [HTTP Request](#): Added Follow All Redirects option.
- [Salesforce](#): Added Record Type ID field.

## Core Functionality

- Fixed UI lag when editing large workflows.

## Bug fixes

- [Nextcloud](#): Fixed issue where List operation on an empty Folder returned an error.
- [Spotify](#): Fixed issues with pagination and infinite executions.

## Contributors

[Jacob Burrell](#), [Лебедев Иван](#)

## n8n@0.132.2

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-08-02

## Bug fixes

- [Interval](#): Fixed issue with infinite executions.

## Contributors

Лебедев Иван

## n8n@0.132.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-08-02

### Core Functionality

- Changed TypeORM version to 0.2.34

## n8n@0.132.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-08-01

### New nodes

- [Freshworks CRM](#)
- [Google Perspective](#)
- [Marketstack](#)
- [NocoDB](#)

### Enhanced nodes

- [Facebook Trigger](#): Added Fields parameter.
- [Gmail](#): Added Sender Name parameter.
- [Home Assistant](#): Added Event resource.
- [Pipedrive](#): Added Deal Product resource.
- [Salesforce](#): Added Document resource with Upload operation.
- [WooCommerce](#): Added Customer resource.

### Core Functionality

- Fixed an issue for large internal values.

### Contributors

[Ed Linklater](#), [Rodrigo Correia](#)

## n8n@0.131.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-07-24

### New nodes

- [Webex by Cisco](#)
- [Webex by Cisco Trigger](#)

## **Enhanced nodes**

- [Pipedrive](#): Added Lead resource. Added Search operation to Organization resource.
- [Taiga Trigger](#): Added Resource and Operations filters.

## **Core Functionality**

- Added Continue-on-fail support to all nodes.
- Added new version notifications.
- Added Refresh List for remote options lists.
- Added `$position` expression variable to return the index of an item within a list.

## **Bug fixes**

- [Spreadsheet File](#): Fixed issue when saving dates.

## **Contributors**

[Anthr@x](#), [Felipe Cecagno](#)

# **n8n@0.130.0**

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-07-18

## **New nodes**

- [AWS DynamoDB](#)
- [Elasticsearch](#)
- [ServiceNow](#)

## **Enhanced nodes**

- [Kafka Trigger](#): Added Read Messages From Beginning option.
- [Salesforce](#): Added Sandbox Environment Type for OAuth2 credentials.
- [Taiga](#): Added Epic, Task, and User Story operations.
- [TheHive](#): Added Custom Fields option to the available Additional Fields.

## **Core Functionality**

- Fixed an issue where failed workflows were displayed as “running”.
- Fixes issues with uncaught errors.

## **Bug fixes**

- [Notion](#): Fixed issue when filtering field data type.

## **Contributors**

[Michael Hirschler](#), [Mika Luhta](#), [Pierre Lanvin](#)

## n8n@0.129.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-07-12

### New nodes

- [Baserow](#)

### Bug fixes

- [SSH](#): Fixed issue with access rights when downloading files.

### Contributors

[Jérémie Pardou-Piquemal](#)

## n8n@0.128.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-07-11

### New nodes

- [Home Assistant](#)
- [Stripe](#)

### Enhanced nodes

- [HTTP Request](#): Added support for arrays in Querystring. Any parameter appearing multiple times with the same name is grouped into an array.
- [Mautic](#): Added Contact Segment resource.
- [Telegram](#): Added Delete operation to the Message resource.

### Core Functionality

- Performance improvement for loading of historical executions (> 3mil) when using Postgres.
- Fixed error handling for unending workflows and display of “unknown” workflow status.
- Fixed format of Workflow ID when downloading from UI Editor to enable compatibility with importing from CLI.

### Bug fixes

- [Microsoft SQL](#): Fixed an issue with sending the connectionTimeout parameter, and creating and updating data using columns with spaces.

## Contributors

[Kaito Udagawa](#), [Rodrigo Correia](#)

## n8n@0.127.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-07-04

### Enhanced nodes

- [Airtable](#): Added Bulk Size option to all Operations.
- [Box](#): Added Share operation to File and Folder resources.
- [Salesforce](#): Added Last Name field to Update operation on Contact resource.
- [Zoho CRM](#): Added Account, Contact, Deal, Invoice, Product, Purchase, Quote, Sales Order, and Vendor resources.

### Core Functionality

- Added a workflow testing framework using a new CLI command to execute all desired workflows. Run `n8n executeBatch --help` for details.
- Added support to display binary video content in Editor UI.

### Bug fixes

- [Google Sheets](#): Fixed an issue with handling 0 value that resulted in empty cells.
- [SSH](#): Fixed an issue with setting passphrases.

## Contributors

[flybluewolf](#), [Kaito Udagawa](#)

## n8n@0.126.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-06-29

### Core Functionality

- Fixed issues with keyboard shortcuts when a modal was open.

### Bug fixes

- [Microsoft SQL](#): Fixed an issue with handling of Boolean values when inserting.
- [Pipedrive](#): Fixed an issue with the node icon.

## n8n@0.126.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-06-27

## New nodes

- [Action Network](#)
- [Google Docs](#)

## Enhanced nodes

- [AWS S3](#): Added Delete operation to the Bucket Resource.
- [Google Analytics](#): Added Dimension Filters to the available Additional Fields.
- [HTTP Request](#): Added Split Into Items option.
- [MQTT](#): Added mqqtts protocol for MQTT credentials.
- [QuickBooks](#): Added Purchase resource with Get and Get All operations.

## Core Functionality

- Templates from the [n8n Workflows](#) page can now be directly imported by appending /workflows/templates/<templateId> to your instance base URL. For example, localhost:5678/workflows/templates/1142.
- Added new Editor UI shortcuts. See [Keyboard Shortcuts](#) for details.
- Fixed an issue causing console errors when deleting a node from the canvas.

## Bug fixes

- [Ghost](#): Fixed an issue with the Get All operation functionality.
- [Google Analytics](#): Fixed an issue that caused an error when attempting to sort with no data present.
- [Microsoft SQL](#): Fixed an issue when escaping single quotes and mapping empty fields.
- [Notion](#): Fixed an issue with pagination of databases and users.

## Contributors

[calvintwr](#), [Jan Baykara](#)

# n8n@0.125.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-06-20

## Enhanced nodes

- [Spotify](#): Added Search operation to Album, Artist, Playlist, and Track resources, and Resume and Volume operations to Player resource.

## Core Functionality

- Implemented new design of the Nodes Panel, adding categories and subcategories, along with improved search. For full details, see the [commits](#).

## Bug fixes

- [MySQL](#): Fixed an issue where n8n was unable to save data due to collation, resulting in workflows ending with Unknown status.

## Contributors

[Amudhan Manivasagam](#), [Carlos Alexandro Becker](#), [Kaito Udagawa](#)

## n8n@0.124.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-06-16

## Core Functionality

- Improved error log messages
- Fixed an issue where the tags got removed when deactivating the workflow or updating settings
- Removed the circular references for the error caused by the request library

## n8n@0.124.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-06-13

## Enhanced nodes

- [Google Drive](#): Added APP Properties and Properties options to the Upload operation of the File resource
- [HTTP Request](#): Added the functionality to log the request to the browser console for testing
- [Notion](#): Added the Include Time parameter date field types
- [Salesforce](#): Added Upsert operation to Account, Contact, Custom Object, Lead, and Opportunity resources
- [Todoist](#): Added the Description option to the Task resource

## Core Functionality

- Implemented the functionality to display the error details in a toast message for trigger nodes
- Improved error handling by removing circular references from API errors

## Bug fixes

- [Jira](#): Fixed an issues with the API version and fixed an issue with fetching the custom fields for the Issue resource

## Contributors

[Jean M](#), [romaincolombo-daily](#), [Thomas Jost](#), [Vincent](#)

## n8n@0.123.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-06-06

### Core Functionality

- Fixed a build issue for missing node icons

## n8n@0.123.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-06-06

### New nodes

- [Git](#)
- [Microsoft To Do](#)

### Enhanced nodes

- [Pipedrive](#): Added a feature to fetch data from the Pipedrive API, added Search operation to the Deals resource, and added custom fields option
- [Spotify](#): Added My Data resource

### Core Functionality

- Fixed issues with NodeViewNew navigation handling
- Fixed an issue with the view crashing with large requests

### Bug fixes

- [AWS Transcribe](#): Fixed issues with options

## Contributors

[Rodrigo Correia](#), [Sam Roquitte](#)

## n8n@0.122.3

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-06-04

## Core Functionality

- Fixed error messages for the text area field
- Added the missing `winston` dependency
- Fixed an issue with adding values using the Variable selector. The deleted values don't reappear
- Fixed an issue with the Error Workflows not getting executed in the queue mode

## Bug fixes

- Notion: Fixed an issue with parsing the last edited time

# n8n@0.122.2

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-05-31

## Enhanced nodes

- Function: Added `console.log` support for writing to browser console
- Function Item: Added `console.log` support for writing to browser console

## Core Functionality

- Fixed an issue that enables clicks on tags
- Fixed an issue with escaping workflow name
- Fixed an issue with selecting variables in the Expression Editor

# n8n@0.122.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-05-30

## Core Functionality

- Fixed an issue with the order in migration rollback

# n8n@0.122.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-05-30

## New nodes

- [AWS Transcribe](#)
- [SSH](#)
- [UptimeRobot](#)

## Enhanced nodes

- [DeepL](#): Added support for Free API
- [Function](#): Added the functionality to log console.log messages to the browser console
- [Function Item](#): Added the functionality to log console.log messages to the browser console

## Core Functionality

- Changed bcrypt library from @node-rs/bcrypt to bcryptjs
- Fixed an issue with optional parameters that have the same name
- Added the functionality to tag workflows
- Fixed errors in the Expression Editor
- Fixed an issue with nodes that only get connected to the second input. This solves the issue of copying and pasting the workflows where only one output of the IF node gets connected to a node

## Bug fixes

- [Google Drive](#): Fixed an issue with the Drive resource
- [Notion](#): Fixed an issue with the filtering fields type and fixed an issue with the link option
- [Switch](#): Fixed an issue with the Expression mode

## Contributors

[Alexander Mustafin](#)

## n8n@0.121.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-06-01

## Core Functionality

- Fixed an issue with copying the output values
- Fixed issues with the Expression Editor
- Made improvements to the Expression Editor

## n8n@0.121.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-05-20

## New nodes

- [Notion](#)

- [Notion Trigger](#)

## Enhanced nodes

- [GraphQL](#): Added Header Auth authentication method
- [Twilio](#): Added API Key authentication method

## Bug fixes

- [HubSpot](#): Fixed an issue with pagination for Deals resource
- [Keap](#): Fixed an issue with the data type of the Order Title field
- Orbit: Fixed an issue with the activity type in Post operation
- [Slack](#): Fixed an issue with the Get Profile operation
- [Strava](#): Fixed an issue with the paging parameter

## Contributors

[Jacob Spizziri](#)

## n8n@0.120.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-05-17

## New nodes

- [iCalendar](#)

## Enhanced nodes

- [Google Cloud Firestore](#): Added the functionality for GeoPoint parsing and added ISO-8601 format for date validation
- [IMAP Email](#): Added the Force reconnect option
- [Paddle](#): Added the Use Sandbox environment API parameter
- [Spotify](#): Added the Position parameter to the Add operation of the Playlist resource
- [WooCommerce](#): Added the Include Credentials in Query parameter

## Core Functionality

- Added await to hooks to fix issues with the Unknown status of the workflows
- Changed the data type of the credentials\_entity field for MySQL database to fix issues with long credentials
- Fixed an issue with the ordering of the executions when the list is auto-refreshed
- Added the functionality that allows reading sibling parameters

## Bug fixes

- [Clockify Trigger](#): Fixed an issue that occurred when the node returned an empty array
- [Google Cloud Firestore](#): Fixed an issue with parsing empty document, and an issue with the detection of date

- [HubSpot](#): Fixed an issue with the Return All option

## Contributors

DeskYT, Daniel Lazaro, [DerEnderKeks](#), [mdasmendel](#)

## n8n@0.119.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-05-09

### Enhanced nodes

- [AWS Comprehend](#): Added the Detect Entities operation
- [AWS Lambda](#): Added the ability to list functions recursively if the number of functions exceeds 50
- [Google Analytics](#): Added pagination to the Report resource
- [Mailjet](#): Added Reply To parameter
- [Redis](#): Added the Increment operation
- [Spreadsheet File](#): Added the Header Row option
- [Webflow Trigger](#): Added Collection Item Created, Collection Item Updated, and Collection Item Deleted events

### Core Functionality

- Implemented timeout for subworkflows
- Removed the deregistration webhooks functionality from the webhook process

### Bug fixes

- [Google Cloud Firestore](#): Fixed an issue with parsing null value
- [Google Sheets](#): Fixed an issue with the Key Row parameter
- [HubSpot](#): Fixed an issue with the authentication

## Contributors

[Nikita](#)

## n8n@0.118.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-05-05

### Core Functionality

- Fixed an issue with error workflows

## n8n@0.118.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-05-02

## New nodes

- [Kitemaker](#)
- [MQTT](#)

## Enhanced nodes

- [CreateDB](#): Added query parameters. The Execute Query operation returns the result from all queries executed instead of just one of the results.
- [ERPNext](#): Added support for self-hosted ERPNext instances
- [FTP](#): Added the functionality to delete folders
- [Google Calendar](#): Added the Continue on Fail functionality
- [Google Drive](#): Added the functionality to add file name when downloading files
- [Gmail](#): Added functionality to handle multiple binary properties
- [Microsoft Outlook](#): Added Is Read and Move option to the Message resource
- [Postgres](#): Added query parameters. The Execute Query operation returns the result from all queries executed instead of just one of the results.
- [QuestDB](#): Added query parameters. The Execute Query operation returns the result from all queries executed instead of just one of the results.
- [QuickBase](#): Added option to use Field IDs
- [TimescaleDB](#): Added query parameters. The Execute Query operation returns the result from all queries executed instead of just one of the results.
- [Twist](#): Added Get, Get All, Delete, and Update operations to the Message Conversation resource. Added Archive, Unarchive, and Delete operations to the Channel resource. Added Thread and Comment resource

## Core Functionality

- Implemented the native `fs/promises` library where possible
- Added the functionality to output logs to the console or a file
- We have updated the minimum required version for Node.js to v14.15. For more details, check out the entry in the [breaking changes](#) page

## Bug fixes

- [GetResponse Trigger](#): Fixed an issue with error handling
- [GitHub Trigger](#): Fixed an issue with error handling
- [GitLab Trigger](#): Fixed an issue with error handling
- [Google Sheets](#): Fixed an issue with the Lookup operation for returning empty rows
- [Orbit](#): Fixed issues with the Post resource
- [Redis](#): Fixed an issue with the node not returning an error
- [Xero](#): Fixed an issue with the Create operation for the Contact resource

## Contributors

[Gustavo Arjones](#), [lublak](#), [Colton Anglin](#), [Mika Luhta](#)

## n8n@0.117.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-04-24

### New nodes

- [Mailcheck](#)
- [n8n Trigger](#)
- [Workflow Trigger](#)

### Enhanced nodes

- [CreateDB](#): Added the Mode option that allows you to execute queries as transactions
- [Nextcloud](#): Added Delete, Get, Get All, and Update operation to the User resource
- [Postgres](#): Added the Mode option that allows you to execute queries as transactions
- [QuestDB](#): Added the Mode option that allows you to execute queries as transactions
- [Salesforce](#): Added Owner option to the Case and Lead resources. Added custom fields to Create and Update operations of the Case resource
- [Sentry.io](#): Added Delete and Update operations to Project, Release, and Team resources
- [TimescaleDB](#): Added the Mode option that allows you to execute queries as transactions
- [Zendesk Trigger](#): Added support to retrieve custom fields

### Core Functionality

- The Activation Trigger node has been deprecated. It has been replaced by two new nodes - the n8n Trigger and the Workflow Trigger node. For more details, check out the entry in the [breaking changes](#) page
- Added the functionality to open the New Credentials dropdown by default

### Bug fixes

- [Google Sheets](#): Fixed an issue with the Lookup operation for returning multiple empty rows
- [Intercom](#): Fixed an issue with the User operation in the Company resource
- [Mautic](#): Fixed an issue with sending the lastActive parameter

## Contributors

[Bart Vollebregt](#), [Ivan Timoshenko](#), [Konstantin Nosov](#), [lublak](#), [Umair Kamran](#),

## n8n@0.116.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-04-20

### Core Functionality

- Fixed a timeout issue with the workflows in the main process

## n8n@0.116.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-04-17

### New nodes

- [Google BigQuery](#)
- [Webflow](#)

### Enhanced nodes

- [Date & Time](#): Added Calculate a Date action that allows you to add or subtract time from a date
- [GitLab](#): Added Get, Get All, Update, and Delete operations to the Release resource
- [Microsoft OneDrive](#): Added Delete operation to the Folder resource
- [Monday](#): Added support for OAuth2 authentication
- [MongoDB](#): Added Limit, Skip, and Sort options to the Find operation and added Upsert parameter to the Update operation. Added the functionality to close the connection after use
- [MySQL](#): Added support for insert modifiers and added support for SSL
- [RabbitMQ](#): Added the functionality to close the connection after use and added support for AMPQS

### Core Functionality

- Changed bcrypt library from bcryptjs to @node-rs/bcrypt
- Improved node error handling. Status codes and error messages in API responses have been standardized
- Added global timeout setting for all HTTP requests (except HTTP Request node)
- Implemented timeout for workers and corrected timeout for sub workflows

### Bug fixes

- [AWS SQS](#): Fixed an issue with API version and casing
- [IMAP](#): Fixed re-connection issue
- [Keap](#): Fixed an issue with the Opt In Reason parameter
- [Salesforce](#): Fixed an issue with loading custom fields

## Contributors

[Allan Daemon](#), [Anton Romanov](#), [Bart Vollebregt](#), [Cassiano Vailati](#),  
[entrailz](#), [Konstantin Nosov](#), [LongYinan](#)

## n8n@0.115.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-04-10

### New nodes

- [Google Slides](#)

### Enhanced nodes

- [GitHub](#): Added Release resource
- [TheHive](#): Added support to fetch observable data types
- [RabbitMQ](#): Added header parameters

### Core Functionality

- Fixed an issue with expressions not being displayed in read-only mode
- Fixed an issue that didn't allow editing JavaScript code in read-only mode
- Added support for configuring the maximum payload size
- Added support to dynamically add menu items

### Bug fixes

- [Jira](#): Fixed an issue with loading issue types with classic project type
- [RabbitMQ Trigger](#): Fixed an issue with the node reusing the same item
- [SendGrid](#): Fixed an issue with the dynamic field generation

## Contributors

[Mika Luhta](#), [Loran](#), [stwonary](#)

## n8n@0.114.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-04-03

### New nodes

- [AWS SQS](#)
- [Copper](#)
- [ERPNext](#)
- [Oura](#)

## Enhanced nodes

- [Google Drive](#): Added support for creating folders for shared drives
- [Google Sheets](#): Added Create and Remove operation to the Sheet resource
- [Harvest](#): Added Update operation to the Task resource
- [Jira](#): Added Reporter field to the Issue resource
- [Postgres](#): Added support for type casting

## Core Functionality

- Fixed an issue with the Redis connection to prevent memory leaks

## Bug fixes

- [Bitwarden](#): Fixed an issue with the Update operation of the Group resource
- [Cortex](#): Fixed an issue where only the last item got returned
- [Invoice Ninja](#): Fixed an issue with the Project parameter
- [Salesforce](#): Fixed an issue with the Get All operation of the Custom Object resource

## Contributors

[Agata M](#), [Allan Daemon](#), [Craig McElroy](#), [mjysci](#)

## n8n@0.113.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-03-26

- New nodes
- Activation Trigger
- Plivo
- Enhanced nodes
- ClickUp: Added Space Tag, Task List, and Task Tag resource
- GitHub: Added pagination to Get Issues and Get Repositories operations
- Mattermost: Added Reaction resource and Post Ephemeral operation
- Move Binary Data: Added Encoding and Add BOM option to JSON to Binary mode and Strip BOM to Binary to JSON mode
- SendGrid: Added Mail resource
- Spotify: Added Library resource
- Telegram: Added Answer Inline Query operation to the Callback resource
- uProc: Added Get ASIN code by EAN code, Get EAN code by ASIN code, Get Email by Social Profile, Get Email by Full name and Company's domain, and Get Email by Full name and Company's name operations
- Bug fixes
- Clearbit: Fixed an issue with the autocomplete URI
- Dropbox: Fixed an issue with the Dropbox credentials by adding the APP Access Type parameter in the credentials. For more details, check out the entry in the [breaking changes](#) page
- Spotify: Fixed an issue with the Delete operation of the Playlist

- resource
- The variable selector now displays empty arrays
  - Fixed a permission issue with the Raspberry Pi Docker image

## n8n@0.112.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-03-19

- New nodes
- DeepL
- Enhanced nodes
- TheHive: Added Mark as Read and Mark as Unread operations and added Ignore SSL Issues parameter to the credentials
- Bug fixes
- AWS SES: Fixed an issue to map CC addresses correctly
- Salesforce: Fixed an issue with custom object for Get All operations and fixed an issue with the first name field for the Create and Update operations for the Lead resource
- Strava: Fixed an issue with the access tokens not getting refreshed
- TheHive: Fixed an issue with the case resolution status
- Fixed an issue with importing separate decrypted credentials
- Fixed issues with the sub-workflows not finishing
- Fixed an issue with the sub-workflows running on the main process
- Fixed concurrency issues with sub-workflows

## n8n@0.111.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-03-12

- New nodes
- Autopilot
- Autopilot Trigger
- Wise
- Wise Trigger
- Enhanced nodes
- Box: Added Get operation to the Folder resource
- Dropbox: Added Search operation to the File resource. All operations are now performed relative to the user's root directory. For more details, check out the entry in the [breaking changes](#) page
- Facebook Graph API: Added new API versions
- Google Drive: Added Update operation to the File resource
- HubSpot: Added the Deal Description option
- Kafka: Added the SASL mechanism
- Monday.com: Added Move operation to Board Item resource
- MongoDB: Added Date field to the Insert and Update operations
- Microsoft SQL: Added connection timeout parameter to credentials
- Salesforce: Added Mobile Phone field to the Lead resource
- Spotify: Added Create a Playlist operation to Playlist resource and Get New Releases to the Album resource
- Bug fixes
- Airtable: Fixed a bug with updating and deleting records
- Added the functionality to expose metrics to Prometheus. Read more about that [here](#)
- Updated fallback values to match the value type

- Added the functionality to display debugging information for pending workflows on exit
- Fixed an issue with queue mode for the executions that shouldn't be saved
- Fixed an issue with workflows crashing and displaying Unknown status in the execution list
- Fixed an issue to prevent crashing while saving execution data when the data field has over 64KB in MySQL
- Updated jws-rsa to version 1.12.1

## n8n@0.110.3

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-03-04

- Bug fixes
- APITemplate.io: Fixed an issue with the naming of the node

## n8n@0.110.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-03-04

- New nodes
- APITemplate.io
- Bubble
- Lemlist
- Lemlist Trigger
- Enhanced nodes
- Microsoft Teams: Added option to reply to a message
- Bug fixes
- Dropbox: Fixed an issue with parsing the response with the Upload operation
- Gmail: Fixed an issue with the scope for the Service Account authentication method and fixed an issue with the label filter
- Google Drive: Fixed an issue with the missing Parent ID field for the Create operation and fixed an issue with the Permissions field
- HelpScout: Fixed an issue with sending tags when creating a conversation
- HTTP Request: Fixed an issue with the raw data and file response
- HubSpot: Fixed an issue with the OAuth2 credentials
- Added support for Date & Time in the IF node and the Switch node
- Fixed an issue with mouse selection when zooming in or out
- Fixed an issue with current executing workflows when using queues for Postgres
- Fixed naming and description for the N8N\_SKIP\_WEBHOOK\_DEREGISTRATION\_SHUTDOWN environment variable
- Fixed an issue with auto-refresh of the execution list

## n8n@0.109.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-02-22

- New nodes
- Bitwarden
- Emelia
- Emelia Trigger
- GoToWebinar
- Raindrop
- Enhanced nodes
- AWS Rekognition: Added the Detect Text type to the Analyze operation for the Image resource
- Google Calendar: Added RRULE parameter to the Get All operation for the Event resource
- Jira: Added User resource and operations
- Reddit: Added the Search operation for the Post resource
- Telegram: Added the Send Location operation
- Bug fixes
- RocketChat: Fixed error responses
- Fixed the issue which caused the execution history of subworkflows (workflows started using the Execute Workflow node) not to be saved
- Added an option to export the credential data in plain text format using the CLI

## n8n@0.108.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-02-15

- New nodes
- Demio
- PostHog
- QuickBooks
- Enhanced nodes
- Trello: Added Create Checklist Item operation to the Checklist resource
- Webhook: Removed trailing slash in routes and updated logic to select dynamic webhook
- Bug fixes
- Google Drive: Fixed an issue with returning the fields the user selects for the Folder and File resources
- Twitter: Fixed a typo in the description
- Webhook: Fixed logic for static route matching
- Added the functionality to sort the values that you add in the IF node, Rename node, and the Set node
- Added the functionality to optionally save execution data after each node
- Added queue mode to scale workflow execution
- Separated webhook from the core to scale webhook separately
- Fixed an issue with current execution query for unsaved running workflows
- Fixed an issue with the regex that detected node names
- n8n now generates a unified execution ID instead of two separate IDs for currently running and saved executions

## n8n@0.107.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-02-08

- New nodes
- AWS Comprehend
- GetResponse Trigger
- Peekalink
- Stackby
- Enhanced nodes
- AWS SES: Added Custom Verification Email resource
- Microsoft Teams: Added Task resource
- Twitter: Added Delete operation to the Tweet resource
- Bug fixes
- Google Drive: Fixed an issue with the Delete and Share operations
- FileMaker: Fixed an issue with the script list parsing
- Updated Node.js version of Docker images to 14.15
- Added a shortcut CTRL + scroll to zoom

## n8n@0.106.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-02-05

- New nodes
- Reddit
- Tapfiliate
- Enhanced nodes
- Airtable Trigger: Added Download Attachment option
- HubSpot: Added Custom Properties option to the Create and Update operations of the Company resource
- MySQL: Added Connection Timeout parameter to the credentials
- Telegram: Added Pin Chat Message and Unpin Chat Message operations for the Message resource
- Bug fixes
- Typeform: Fixed an issue with the OAuth2 authentication method
- Added support for s and u flags for regex in the IF node and the Switch node

## n8n@0.105.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-02-01

- New nodes
- Discourse
- SecurityScorecard
- TimescaleDB
- Enhanced nodes
- Affinity: Added List and List Entry resource
- Asana: Added Project IDs option to the Create operation of the Task resource
- HubSpot Trigger: Added support for multiple subscriptions
- Jira: Added Issue Attachment resource and added custom fields to Create and Update operations of the Issue resource
- Todoist: Added Section option

- Bug fixes
- SIGNL4: Fixed an issue with the attachment functionality
- Added variable \$mode to check the mode in which the workflow is being executed

## n8n@0.104.2

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-01-27

- Fixed an issue with the credentials parameters that have the same name

## n8n@0.104.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-01-26

- Fixed a bug with expressions in credentials

## n8n@0.104.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-01-26

- New nodes
- Compression
- Enhanced nodes
- GitHub: Added Invite operation to the User resource
- EmailReadImap: Increased the authentication timeout
- Mautic: Added Custom Fields option to the Create and Update operations of the Contact resource. Also, the Mautic OAuth credentials have been updated. Now you don't have to enter the Authorization URL and the Access Token URL
- Nextcloud: Added User resource
- Slack: Added Get Permalink and Delete operations to the Message resource
- Webhook: Added support for request parameters in webhook paths
- Bug fixes
- Google Drive: Fixed the default value for the Send Notification Email option
- Added support for expressions to credentials
- Removed support for MongoDB as a database for n8n. For more details, check out the entry in the [breaking changes](#) page

## n8n@0.103.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-01-21

- Bug fixes

- Trello: Fixed the icon

## n8n@0.103.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-01-21

- New nodes
- SendGrid
- Enhanced nodes
- AMQP: Added Container ID, Reconnect, and Reconnect limit options
- AMQP Trigger: Added Container ID, Reconnect, and Reconnect Limit options
- GitHub: Added Review resource
- Google Drive: Added Drive resource
- Trello: Added Get All and Get Cards operation to the List resource
- Bug fixes
- AWS Lambda: Fixed an issue with signature
- AWS SNS: Fixed an issue with signature
- Fixed an issue with nodes not executing if two input gets passed and one of them didn't return any data
- The code editor doesn't get closed when clicked anywhere outside the editor
- Added CLI commands to [export](#) and [import](#) credentials and workflows
- The title in the browser tab now resets for new workflows

## n8n@0.102.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-01-15

- New nodes
- Beeminder
- Enhanced nodes
- Crypto: Added hash type SHA384
- Google Books: Added support for user impersonation
- Google Drive: Added support for user impersonation
- Google Sheets: Added support for user impersonation
- Gmail: Added support for user impersonation
- Microsoft Outlook: Added support for a shared mailbox
- RabbitMQ: Added Exchange mode
- Salesforce: Added filters to all Get All operations
- Slack: Made changes to the properties As User and Ephemeral. For more details, check out the entry in the [breaking changes](#) page
- Typeform Trigger: The node now displays the recall information in the question in square brackets. For more details, check out the entry in the [breaking changes](#) page
- Zendesk: Removed the Authorization URL and Access Token URL fields from the OAuth2 credentials. The node now uses the subdomain passed by a user to connect to Zendesk.
- Bug fixes
- CoinGecko: Fixed an issue to process multiple input items correctly

## n8n@0.101.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2021-01-07

- New nodes
- Google Analytics
- PhantomBuster
- Enhanced nodes
- AWS: Added support for custom endpoints
- Gmail: Added an option to send messages formatted as HTML
- Philips Hue: Added Room/Group name to Light name to make it easier to identify lights
- Slack: Added ephemeral message option
- Telegram: Removed the Bot resource as the endpoint is no longer supported
- Bug fixes
- E-goi: Fixed the name of the node
- Edit Image: Fixed an issue with the Border operation
- HTTP Request: Fixed batch sizing to work when `batchSize = 1`
- PayPal: Fixed a typo in the Environment field
- Split In Batches: Fixed a typo in the description
- Telegram: Fixed an issue with the Send Audio operation
- Based on your settings, vacuum runs on SQLite on startup
- Updated Axios to version 0.21.1

## n8n@0.100.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-12-30

- New nodes
- Microsoft Outlook
- Enhanced nodes
- ActiveCampaign: The node loads more options for the fields
- Asana: Added Subtask resource and Get All operation for the Task resource
- Edit Image: Added Multi Step operation
- HTTP Request: Added Use Querystring option
- IF: Added Ends With and Starts With operations
- Jira: Added Issue Comment resource
- Switch: Added Ends With and Starts With operations
- Telegram: Added File resource
- Bug fixes
- Box Trigger: Fixed a typo in the description
- Edit Image: Fixed an issue with multiple composite operations
- HTTP Request: Fixed an issue with the binary data getting used by multiple nodes
- S3: Fixed an issue with uploading files
- Stripe Trigger: Fixed an issue with the existing webhooks
- Telegram: Fixed an issue with the Send Audio operation
- Binary data stays visible if a node gets re-executed

## n8n@0.99.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-12-24

- Fixed a bug that caused HTML to render in JSON view

## n8n@0.99.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-12-24

- New nodes
- e-goi
- RabbitMQ
- RabbitMQ Trigger
- uProc
- Enhanced nodes
- ActiveCampaign: Added the functionality to load the tags for a user
- FTP: Added Delete and Rename operation
- Google Cloud Firestore: The node now gives the Collection ID in response
- Iterable: Added User List resource
- MessageBird: Added Balance resource
- TheHive Trigger: Added support for the TheHive3 webhook events, and added Log Updated and Log Deleted events
- Bug fixes
- Dropbox: Fixed an issue with the OAuth credentials
- Google Sheets: Fixed an issue with the parameters getting hidden for other operations
- Added functionality to copy the data and the path from the output
- Fixed an issue with the node getting selected after it was duplicated

## n8n@0.98.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-12-16

- New nodes
- Brandfetch
- Pushcut
- Pushcut Trigger
- Enhanced nodes
- Google Sheets: Added Spreadsheet resource
- IF: Added Is Empty option
- Slack: Added Reaction and User resource, and Member operation to the Channel resource
- Spreadsheet File: Added the option Include Empty Cell to display empty cells

- Webhook: Added option to send a custom response body. The node can now also return string data
- Bug fixes
- GitLab: Fixed an issue with GitLab OAuth credentials. You can now specify your GitLab server to configure the credentials
- Mautic: Fixed an issue with the OAuth credentials
- If a workflow is using the Error Trigger node, by default, the workflow will use itself as the Error Workflow
- Fixed a bug that caused the Editor UI to display an incorrect (save) state upon activating or deactivating a workflow ## n8n@0.97.0  
For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-12-10

- New nodes
- Ghost
- NASA
- Snowflake
- Twist
- Enhanced nodes
- Automizy: Added options to add and remove tabs for the Update operation of the Contact resource
- Pipedrive: Added label field to Person, Organization, and Deal resources. Also added Update operation for the Organization resource
- Bug fixes
- Fixed a bug that caused OAuth1 requests to break
- Fixed Docker user mount path

## n8n@0.96.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-12-03

- New nodes
- Cortex
- Iterable
- Kafka Trigger
- TheHive
- TheHive Trigger
- Yourls
- Enhanced nodes
- HubSpot: Added Contact List resource and Search operation for the Deal resource
- Google Calendar: You can now add multiple attendees in the Attendees field

- Slack: The node now loads both private and public channels
- Bug Fixes
- MQTT: Fixed an issue with the connection. The node now uses `mqtt@4.2.1`
- Fixed a bug which caused the Trigger-Nodes to require data from the first output
- Added configuration to load only specific nodes

## n8n@0.95.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-11-25

- Bug Fixes
- Airtable Trigger: Fixed the icon of the node

## n8n@0.95.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-11-25

- New nodes
- Airtable Trigger
- LingvaNex
- OpenThesaurus
- ProfitWell
- Quick Base
- Spontit
- Enhanced nodes
- Airtable: The Application ID field has been renamed to Base ID, and the Table ID field has been renamed to Table. The List operation now downloads attachments automatically
- Harvest: Moved the account field from the credentials to the node parameters. For more details, check out the entry in the [breaking changes](#) page
- Bug Fixes
- Slack: Fixed an issue with creating channels and inviting users to a channel

## n8n@0.94.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-11-20

- Bug Fixes
- GraphQL: Fixed an issue with the variables
- WooCommerce Trigger: Fixed an issue with the webhook. The node now reuses a webhook if it already exists.

## n8n@0.94.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-11-19

- New nodes
- Google Cloud Natural Language
- Google Firebase Cloud Firestore
- Google Firebase Realtime Database
- Humanotic AI
- Enhanced nodes
- ActiveCampaign: Added Contact List and List resource
- Edit Image: Added support for drawing, font selection, creating a new image, and added the Composite resource
- FTP: Added Private Key and Passphrase fields to the SFTP credentials and made the directory creation more robust
- IMAP: Increased the timeout
- Matrix: Added option to send notice, emote, and HTML messages
- Segment: Made changes to the properties traits and properties. For more details, check out the entry in the [breaking changes](#) page
- Bug Fixes
- GraphQL: Fixed an issue with the variables
- Mailchimp: Fixed an issue with the OAuth credentials. The credentials are now sent with the body instead of the header
- YouTube: Fixed a typo for the Unlisted option
- Added horizontal scrolling

## n8n@0.93.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-11-11

- New nodes
- GetResponse
- Gotify
- Line
- Strapi
- Enhanced nodes
- AMQP: Connection is now closed after a message is sent
- AMQP Trigger: Added Message per Cycle option to retrieve the specified number of messages from the bus for every cycle
- HubSpot: Added Custom Properties for the Deal resource as Additional Fields
- Jira: The node retrieves all the projects for the Project field instead of just 50
- Mattermost: Improved the channel selection
- Microsoft SQL: Added TLS parameter for the credentials
- Pipedrive Trigger: Added OAuth authentication method. For more details, check out the entry in the [breaking changes](#) page
- Segment: Added Custom Traits option for the Traits field
- Bug Fixes
- Shopify Trigger: Fixed an issue with activating the workflow
- For custom nodes, you can now set custom documentation URLs

## n8n@0.92.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-11-04

- New nodes
- Facebook Trigger
- Google Books
- Orbit
- Storyblok
- Enhanced nodes
- Google Drive: Removed duplicate parameters
- Twitter: Added Direct Message resource
- Bug Fixes
- Gmail: Fixed an issue with the encoding for the subject field
- Improved the Editor UI for the save workflow functionality

## n8n@0.91.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-10-23

- New nodes
- Kafka
- MailerLite
- MailerLite Trigger
- Pushbullet
- Enhanced nodes
- Airtable: Added Ignore Fields option for the Update operation
- AMQP Sender: Added Azure Service Bus support
- Google Calendar: Added Calendar resource and an option to add a conference link
- G Suite Admin: Added Group resource
- HTTP Request: Added Batch Size and Batch Interval option
- Mautic: Added Company resource
- Salesforce: Added OAuth 2.0 JWT authentication method
- Bug Fixes
- IF: Fixed an issue with undefined expression
- Paddle: Fixed an issue with the Return All parameter
- Switch: Fixed an issue with undefined expression
- Added CLI commands to deactivate the workflow
- Added an option to get the full execution data from the server
- The Editor UI gives an alert if you redirect without saving a workflow
- The Editor UI now indicates if a workflow is saved or not
- Improved support for touch devices
- Node properties now load on demand
- Updated the Node.js version for the Docker images

## n8n@0.90.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-10-23

- Added a check for the Node.js version on startup. For more details, check out the entry in the [breaking changes](#) page
- Bug Fixes

- Google Translate: Fixed an issue with the rendering of the image in n8n.io

## n8n@0.89.2

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-10-22

- Bug Fixes
- Strava Trigger: Fixed a typo in the node name

## n8n@0.89.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-10-22

- Removed debug messages

## n8n@0.89.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-10-22

- New Nodes
- Pushover
- Strava
- Strava Trigger
- Google Translate
- Bug Fixes
- HTTP Request: Fixed an issue with the POST request method for the 'File' response format
- Fixed issue with displaying non-active workflows as active
- Fixed an issue related to multiple-webhooks

## n8n@0.88.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-10-16

- Bug Fixes
- HTTP Request: Fixed an issue with the Form-Data Multipart and the RAW/Custom Body Content Types

## n8n@0.88.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-10-16

- Enhanced Fixes

- Matrix: Added support for specifying a Matrix Homeserver URL
- Salesforce: Added Custom Object resource and Custom Fields and Sort options
- Bug Fixes
- AWS SES: Fixed an issue with the Send Template operation for the Email resource
- AWS SNS Trigger: Fixed an issue with the Subscriptions topic

## n8n@0.87.2

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-10-15

- Bug Fixes
- Google Sheets: Fixed an issue with spaces in sheet names
- Automizy: Fixed an issue with the default resource

## n8n@0.87.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-10-15

- Bug Fixes
- Gmail: Fixed an issue with the Message ID
- HTTP Request: Fixed an issue with the GET Request
- Added HMAC-SHA512 signature method for OAuth 1.0

## n8n@0.87.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-10-14

- New nodes
- Automizy
- AWS Rekognition
- Matrix
- Sendy
- Vonage
- WeKan
- Enhanced nodes
- AWS SES: Added Send Template operation for the Email resource and added the Template resource
- ClickUp: Added Time Entry and Time Entry Tag resources
- Function: The Function field is now called the JavaScript Code field
- Mailchimp: Added Campaign resource
- Mindee: Added currency to the simplified response
- OneDrive: Added Share operation
- OpenWeatherMap: Added Language parameter
- Pipedrive: Added additional parameters to the Get All operation for the Note resource
- Salesforce: Added Flow resource
- Spreadsheet File: Added Range option for the Read from file operation
- Bug Fixes

- ClickUp Trigger: Fixed issue with creating credentials
- Pipedrive Trigger: Fixed issue with adding multiple webhooks to Pipedrive
- The link.fish Scrape node has been removed from n8n. For more details, check out the entry in the [breaking changes](#) page

## n8n@0.86.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-10-06

- Enhanced nodes
- CoinGecko: Small fixes to the CoinGecko node

## n8n@0.86.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-10-05

- New nodes
- Clockify
- CoinGecko
- G Suite Admin
- Mindee
- Wufoo Trigger
- Enhanced nodes
- Slack: Added User Profile resource
- Mattermost: Added Create and Invite operations for the User resource
- Bug Fixes
- S3: Fixed issue with uploading files
- Webhook ID gets refreshed on node duplication

## n8n@0.85.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-09-30

- Enhanced nodes
- Postgres: Added Schema parameter for the Update operation
- Bug Fixes
- Jira: Fixed a bug with the Issue Type field
- Pipedrive Trigger: Fixed issues with the credentials
- Changed the bcrypt library to `bcrypt.js` to make it compatible with Windows
- The OAuth callback URLs are now generated in the backend

## n8n@0.84.4

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-09-23

- Bug Fixes
- Google Sheets: Fixed issues with the update and append operations

## n8n@0.84.3

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-09-23

- Fixed an issue with the build by setting jwks-rsa to an older version

## n8n@0.84.2

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-09-23

- Fixed an issue with the OAuth window. The OAuth window now closes after authentication is complete

## n8n@0.84.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-09-23

- Additional endpoints can be excluded from authentication checks. Multiple endpoints can be added separated by colons

## n8n@0.84.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-09-23

- Enhanced nodes
- Twitter: Added support for auto mention of users in reply tweets
- Bug Fixes
- Google Sheets: Fixed issue with non-Latin sheet names
- HubSpot: Fixed naming of credentials
- Microsoft: Fixed naming of credentials
- Mandrill: Fixed attachments with JSON parameters
- Expressions now use short variables when selecting input data for the current node
- Fixed issue with renaming credentials for active workflows

## n8n@0.83.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-09-18

- New nodes
- LinkedIn
- Taiga
- Taiga Trigger
- Enhanced nodes
- ActiveCampaign: Added multiple functions, read more [here](#)
- Airtable: Added typecast functionality
- Asana: Added OAuth2 support
- ClickUp: Added OAuth2 support
- Google Drive: Added share operation
- IMAP Email: Added support for custom rules when checking emails
- Sentry.io: Added support for self-hosted version
- Twitter: Added retweet, reply, and like operations
- WordPress: Added author field to the post resource
- Bug Fixes
- Asana Trigger: Webhook validation has been deactivated
- Paddle: Fixed returnData format and coupon description
- The ActiveCampaign node has [breaking changes](#)
- Fixed issues with test-webhook registration

## n8n@0.82.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-09-14

- Speed for basic authentication with hashed password has been improved

## n8n@0.82.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-09-14

- New nodes
- Microsoft Teams
- Enhanced nodes
- Freshdesk: Added Freshdesk contact resource
- HTTP Request: Run parallel requests in HTTP Request Node
- Bug Fixes
- Philips Hue: Added APP ID to Philips Hue node credentials
- Postmark Trigger: Fixed parameters for the node
- The default space between nodes has been increased to two units
- Expression support has been added to the credentials
- Passwords for your n8n instance can now be hashed

## n8n@0.81.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-09-09

- New nodes
- Sentry.io
- Enhanced nodes

- Asana
- ClickUp
- Clockify
- Google Contacts
- Salesforce
- Segment
- Telegram
- Telegram Trigger

## n8n@0.80.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-09-02

- New nodes
- Customer.io
- MQTT Trigger
- S3
- Enhanced nodes
- Acuity Scheduling
- AWS S3
- ClickUp
- FTP
- Telegram Trigger
- Zendesk

## n8n@0.79.3

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-08-30

- The bug that caused the workflows to not get activated correctly has been fixed

## n8n@0.79.2

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-08-28

- Added missing rawBody for “application/x-www-form-urlencoded”

## n8n@0.79.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-08-28

- Enhanced nodes
- Contentful
- HTTP Request
- Postgres
- Webhook

- Removed Test-Webhook also in case checkExists fails
- HTTP Request node doesn't overwrite accept header if it's already set
- Add rawBody to every request so that n8n doesn't give an error if body is missing

## n8n@0.79.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-08-27

- New nodes
- Contentful
- ConvertKit
- ConvertKit Trigger
- Paddle
- Enhanced nodes
- Airtable
- Coda
- Gmail
- HubSpot
- IMAP Email
- Postgres
- Salesforce
- SIGNL4
- Todoist
- Trello
- YouTube
- The Todoist node has [breaking changes](#)
- Added dynamic titles on workflow execution
- Nodes will now display a link to associated credential documentation

## n8n@0.78.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-08-18

- New nodes
- Gmail
- Google Contacts
- Unleashed Software
- YouTube
- Enhanced nodes
- AMQP
- AMQP Trigger
- Bitly
- Function Item
- Google Sheets
- Shopify
- Todoist
- Enhanced support for [JWT based authentication](#)
- Added an option to execute a node once, using data of only the first item

## n8n@0.76.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-08-05

- New nodes
- Customer.io Trigger
- FTP
- Medium
- Philips Hue
- TravisCI
- Twake
- Enhanced nodes
- CrateDB
- Move Binary Data
- Nodes will now display a link to associated documentation

## n8n@0.75.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-07-26

- New nodes
- Box
- Box Trigger
- CrateDB
- Jira Trigger
- Enhanced nodes
- GitLab
- Nextcloud
- Pipedrive
- QuestDB
- Webhooks now support OPTIONS request

## n8n@0.74.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-07-15

- New nodes
- Hacker News
- QuestDB
- Xero
- Enhanced nodes
- Affinity Trigger
- HTTP Request
- Mailchimp
- MongoDB
- Pipedrive
- Postgres
- UpLead
- Webhook
- Webhook URLs are now handled independently of the workflow ID by <https://{{hostname}}/webhook/{{path}}> instead of the older

[https://{{hostname}}/webhook/{{workflow\\_id}}/{{node\\_name}}/{{path}}](https://{{hostname}}/webhook/{{workflow_id}}/{{node_name}}/{{path}}).

## n8n@0.73.1

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-07-08

- Enhanced nodes
- Microsoft SQL

## n8n@0.73.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-07-08

- New nodes
- CircleCI
- Microsoft SQL
- Zoom
- Enhanced nodes
- Postmark Trigger
- Salesforce
- It's now possible to set default values for credentials that get prefilled, and the user can't change.

## n8n@0.72.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-07-02

- Enhanced nodes
- Drift
- Eventbrite Trigger
- Facebook Graph API
- Pipedrive
- Fixed credential issue for the Execute Workflow node

## n8n@0.71.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-06-25

- New nodes
- Google Tasks
- SIGNL4
- Spotify
- Enhanced nodes
- HubSpot
- Mailchimp
- Typeform
- Webflow

- Zendesk
- Added Postgres SSL support
- It's now possible to deploy n8n under a subfolder

## n8n@0.70.0

For a comprehensive list of changes, check out the [commits](#) for this version.

**Release date:** 2020-06-13

- Enhanced nodes
  - GitHub
  - Mautic Trigger
  - Monday.com
  - MongoDB
  - Fixed the issue with multiuser-setup
- 

## n8n v2.0 breaking changes

n8n v2.0 will be released soon. This document highlights important breaking changes and actions you should take to prepare for the transition. These updates improve security, simplify configuration, and remove legacy features.

The release of n8n 2.0 continues n8n's commitment to providing a secure, reliable, and production-ready automation platform. This major version includes important security enhancements and cleanup of deprecated features.

### Behavior changes

#### **Return expected sub-workflow data when the sub-workflow resumes from waiting (waiting for webhook, forms, HITL, etc.)**

Previously, when an execution (parent) called a sub-execution (child) that contained a node that causes the sub-execution to enter the waiting state and the parent-execution is set up to wait for the sub-execution's completion, the parent-execution would receive incorrect results.

Entering the waiting state would happen for example if the sub-execution contains a Wait node with a timeout higher than 65 seconds or a webhook call or a form submission, or a human-in-the-loop node, like the slack node.

Parent-Workflow: [Image: Parent-Workflow]

Sub-Workflow: [Image: Sub-Workflow]

v1: The parent-execution reproduces the sub-execution's input as its output.: [Image: v1: Parent execution won't receive the result of the child execution]

v2: The parent execution receives the result of the child execution:  
[Image: v2: Parent execution will receive the result of the child execution]

This allows using human-in-the-loop nodes in the sub-workflow and use the results (for example approving or declining an action) in the parent-workflow.

**Migration path:** Review any workflows that call sub-workflows and expect to receive the input to the sub-workflow. Update these workflows to handle the new behavior, where the parent-workflow receives the output from the end of the child-workflow instead.

## Removed nodes for retired services

The following nodes have been removed because the external services they connect to are no longer available:

- Spontit node
- crowd.dev node
- Kitemaker node
- Automizy node

**Migration path:** If your workflows use any of these nodes, update or remove those workflows to avoid errors.

## Security

### Block environment variable access from Code Node by default

To improve security, n8n will block access to environment variables from the Code node by default. The default value for `N8N_BLOCK_ENV_ACCESS_IN_NODE` is now set to true.

**Migration path:** If your workflows require access to environment variables in Code nodes, set `N8N_BLOCK_ENV_ACCESS_IN_NODE=false` in your environment configuration. For sensitive data, use credentials or other secure methods instead of environment variables.

### Enforce settings file permissions

n8n will require strict file permissions for configuration files to improve security. By default, configuration files must use `0600` permissions, which means only the file owner can read and write them. This approach is similar to how SSH protects private keys.

**Migration path:** To test this behavior before v2.0, set `N8N_ENFORCE_SETTINGS_FILE_PERMISSIONS=true`. If your environment doesn't support file permissions (for example, on Windows), set `N8N_ENFORCE_SETTINGS_FILE_PERMISSIONS=false` to disable this requirement.

### Enable task runners by default

n8n will enable task runners by default to improve security and isolation. All Code node executions will run on task runners.

**Migration path:** Before upgrading to v2.0, set `N8N_RUNNERS_ENABLED=true` to test this behavior. Make sure your infrastructure meets the requirements for running task runners. For additional security, consider using [external mode](#).

## Remove task runner from n8nio/n8n docker image

Starting with v2.0, the main n8nio/n8n Docker image will no longer include the task runner for external mode. You must use the separate n8nio/runners Docker image to run task runners in external mode.

**Migration path:** If you run task runners in Docker with external mode, update your setup to use the n8nio/runners image instead of n8nio/n8n.

## Remove Pyodide-based Python Code node and tool

n8n will remove the Pyodide-based Python Code node and tool and replace them with a [task runner-based](#) implementation that uses native Python for better security and performance. Starting in v2.0, you can only use Python Code nodes with task runners in [external mode](#) and native Python tools.

The native Python Code node doesn't support built-in variables like `_input` or dot access notation, which were available in the Pyodide-based version. For details, see the [Code node documentation](#).

The native Python tool supports `_query` for the input string that the AI Agent passes to the tool when it calls it.

**Migration path:** To continue using Python in Code nodes, set up task runners in external mode and review your existing Python Code nodes and tools for compatibility.

## Disable ExecuteCommand and LocalFileTrigger nodes by default

n8n will disable the ExecuteCommand and LocalFileTrigger nodes by default because they pose security risks. These nodes allow users to run arbitrary commands and access the file system.

**Migration path:** If you need to use these nodes, remove them from the disabled nodes list in your n8n configuration by updating the `NODES_EXCLUDE` environment variable. For example, set `NODES_EXCLUDE="[]"` to enable all nodes, or remove only the specific nodes you need.

## Require authentication on OAuth callback URLs by default

n8n will require authentication for OAuth callback endpoints by default. The default value for `N8N_SKIP_AUTH_ON_OAUTH_CALLBACK` will change from `true` (no authentication required) to `false` (authentication required).

**Migration path:** Before upgrading to v2.0, set `N8N_SKIP_AUTH_ON_OAUTH_CALLBACK=false` and test your OAuth integrations to ensure they work with authentication enabled.

## **Set default value for N8N\_RESTRICT\_FILE\_ACCESS\_TO**

n8n will set a default value for N8N\_RESTRICT\_FILE\_ACCESS\_TO to control where file operations can occur. This affects the ReadWriteFile and ReadBinaryFiles nodes. By default, these nodes can only access files in the `~/n8n-files` directory.

**Migration path:** Review your workflows that use file nodes and make sure they only access files in the allowed directory. If you need to allow access to other directories, set the N8N\_RESTRICT\_FILE\_ACCESS\_TO environment variable to your desired path.

## **Change the default value of N8N\_GIT\_NODE\_DISABLE\_BARE\_REPOS to true**

By default, the Git node will now block bare repositories for security reasons. The default value for N8N\_GIT\_NODE\_DISABLE\_BARE\_REPOS is set to `true`, which means bare repositories are disabled unless you change this setting.

**Migration path:** If your workflows need to use bare repositories, set `N8N_GIT_NODE_DISABLE_BARE_REPOS=false` in your environment configuration to enable them.

## **Data**

### **Drop MySQL/MariaDB support**

n8n will no longer support MySQL and MariaDB as storage backends. This support was deprecated in v1.0. For best compatibility and long-term support, use PostgreSQL. MySQL node will continue to be supported as before.

**Migration path:** Before upgrading to v2.0, use the database migration tool to move your data from MySQL or MariaDB to PostgreSQL or SQLite.

### **Remove SQLite legacy driver**

n8n will remove the legacy SQLite driver due to reliability issues. The pooling driver will become the default and only SQLite driver. The pooling driver uses WAL mode, a single write connection, and a pool of read connections. Our benchmarks show it can be up to 10 times faster.

**Migration path:** The `sqlite-pooled` driver will become the default automatically. You can enable pooling now by setting `DB_SQLITE_POOL_SIZE` to a value greater than 0. The default pool size will be set to 2.

### **Remove in-memory binary data mode**

n8n will remove the default mode for N8N\_DEFAULT\_BINARY\_DATA\_MODE, which keeps execution binary data in memory during execution. For better performance and stability the following options will be available starting from v2:

- `filesystem`: Binary data is stored in the filesystem. Default option in regular mode.
- `database`: Binary data is stored in the database. Default option in queue mode.
- `s3`: Binary data is stored in S3 compatible store.

The `N8N_AVAILABLE_BINARY_DATA_MODES` setting will also be removed, so the mode is now determined only by `N8N_DEFAULT_BINARY_DATA_MODE`.

**Migration path:** Filesystem or database mode will be used automatically based on configuration. Make sure your n8n instance has enough disk space to store binary data. For details, see the [binary data configuration](#).

## Configuration & Environment

### Upgrade dotenv

n8n loads environment configuration from a `.env` file using the `dotenv` library. The library will be upgraded from version 8.6.0 to the latest version, which may change how `.env` files are parsed. Key breaking changes include:

- Backtick support ([#615](#)): If your values contain backticks, wrap them in single or double quotes.
- Multiline support: You can now use multiline values.
- `#` marks the beginning of a comment: Lines starting with `#` are treated as comments.

**Migration path:** Review the [dotenv changelog](#) and update your `.env` file to ensure compatibility with the new version.

### Remove n8n --tunnel option

The `n8n --tunnel` command-line option will be removed in v2.0.

**Migration path:** If you currently use the `--tunnel` option for development or testing, switch to an alternative tunneling solution such as ngrok, localtunnel, or Cloudflare Tunnel. Update your workflow and documentation to reflect this change.

### Remove QUEUE\_WORKER\_MAX\_STALLED\_COUNT

The `QUEUE_WORKER_MAX_STALLED_COUNT` environment variable and the Bull retry mechanism for stalled jobs will be removed because they often caused confusion and didn't work reliably.

**Migration path:** Delete this environment variable from your configuration. After upgrading, n8n will no longer automatically retry stalled jobs. If you need to handle stalled jobs, consider implementing your own retry logic or monitoring.

## CLI & Workflow

### Replace CLI command `update:workflow`

The `update:workflow` CLI command will be deprecated and replaced by two new commands to deliver similar functionality and more clarity:

- `publish:workflow` with parameters `id` and `versionId` (optional)
  - The `--all` parameter will be removed to prevent accidental publishing of workflows in production environments
- `unpublish:workflow` with parameters `id` and `all`

**Migration path:** Use the new `publish:workflow` command to publish workflows individually by ID, optionally specifying a version. For unpublishing, use the new `unpublish:workflow` command. This provides better clarity and control over workflow publishing states.

## External Hooks

### Deprecated frontend workflow hooks

The hooks `workflow.activeChange` and `workflow.activeChangeCurrent` will be deprecated. These will be replaced by a new hook `workflow.published`. The new hook will be triggered when any version of a workflow is published.

**Migration path:** Update your code to use the new `workflow.published` hook instead of `workflow.activeChange` and `workflow.activeChangeCurrent`. This hook provides more consistent behavior and will be triggered whenever a workflow version is published.

## Release channels

n8n has renamed the release channels from `latest` and `next` to `stable` and `beta`, respectively.

The `stable` tag designates the latest stable release, and the `beta` tag designates the latest experimental release. These tags are available on both npm and Docker Hub. For now, n8n will continue to tag releases as `latest` and `next`. These tags will be removed in a future major version.

**Recommendation:** Pin your n8n version to a specific version number, for example, `2.0.0`.

## Reporting issues

If you run into any problems while updating to n8n 2.0, visit the community [forum](#) for help and support.

---

## n8n v2.0 migration tool

The migration tool helps you prepare your n8n instance for upgrading to version 2.0 by identifying workflows and configurations that need attention before the upgrade.

[Image: Migration tool]

You can see all breaking changes for v2 [on this page](#).

## Accessing the Tool

Navigate to **Settings > Migration Report** to view your compatibility status.

## Understanding Your Migration Status

At the top of the page, you'll see:

"X out of Y workflows are compatible with n8n 2.0"

This tells you how many workflows will continue working without changes after upgrading. Your goal is to address the issues preventing the remaining workflows from being compatible, as well as global instance issues.

## Viewing Issues

The tool organizes potential problems into two categories:

### Workflow Issues Tab

Shows breaking changes that affect specific workflows in your instance. What you'll see for each issue:

- **Issue title:** A clear name for the problem
- **Severity badge (Critical/Medium/Low):** How urgent this is to fix
  - **Critical:** Fix before upgrading or workflows will fail
  - **Medium:** May cause unexpected behavior or require attention soon
  - **Low:** Minor changes or deprecations that won't break functionality
- **Description:** Explanation of what's changing and why it matters
- **Documentation link:** Click to read detailed migration explanations
- **Affected workflow count:** How many of your workflows have this issue

### Workflow Issue Detail Page

Click **X workflows affected** to see all affected workflows. What you'll see for each workflow:

- **Name:** The workflow name. Click on the name to open the workflow editor.
- **State:** Whether workflow is published or not
- **Node affected:** The list of all the workflow nodes affected by the issue. You can click on each to open the workflow editor with the specific node view opened.
- **Number of executions:** The total number of executions of the workflow
- **Last executed:** The date the workflow was last executed
- **Last updated:** The date the workflow was last updated

## Instance Issues Tab

Shows configuration changes that affect your entire n8n instance, not specific workflows. What you'll see for each issue:

- Same information as workflow issues (title, severity, description, docs)
- **No workflow count:** These are global settings that apply instance-wide

The v2.0 migration tool scans your n8n instance to identify potential compatibility issues and configuration changes required for upgrading to v2.0. This reference details each check the tool performs, explains the impact of detected issues, and provides recommendations to prepare your instance for migration.

## Understanding Empty States

### No Workflow Issues Found

All your workflows are compatible with v2.0. Check the **Instance Issues** tab to ensure your server configuration is also ready.

### No Instance Issues Found

Your instance configuration is compatible with v2.0. Check the **Workflow Issues** tab to ensure all workflows are also ready.

### Both Tabs Empty

Your n8n instance is fully ready to upgrade to version 2.0.

## Recommended Workflow

### Initial Assessment

- Review the compatibility summary
- Browse all issues in both tabs to understand the scope

### Sort by Severity

- Start with Critical issues (they'll break workflows)
- Move to Medium issues (may cause problems)
- Address Low issues last (deprecation warnings)

### Fix Workflow Issues

- Click into each issue to see affected workflows
- Read the documentation for fix instructions
- Update each workflow as needed
- Test workflows in a development environment

### Address Instance Issues

- Update environment variables or server configuration
- Follow documentation for each instance-level change

## Verify Your Work

- Click **Refresh** to re-scan. If you don't see any **Refresh** button, just reload the page to re-scan.
- Confirm that unresolved issues don't remain
- Verify compatibility count matches total workflows

## Proceed with Upgrade

After addressing all issues, you're ready to upgrade to n8n 2.0

---

# n8n v1.0 migration guide

This document provides a summary of what you should be aware of before updating to version 1.0 of n8n.

The release of n8n 1.0 marks a milestone in n8n's journey to make n8n available for demanding production environments. Version 1.0 represents the hard work invested over the last four years to make n8n the most accessible, powerful, and versatile automation tool. n8n 1.0 is now ready for use in production.

## New features

### Python support in the Code node

Although JavaScript remains the default language, you can now also select Python as an option in the Code node and even make use of many Python modules. Note that Python is unavailable in Code nodes added to a workflow before v1.0.

[PR #4295](#), [PR #6209](#)

### Execution order

n8n 1.0 introduces a new execution order for multi-branch workflows:

In multi-branch workflows, n8n needs to determine the order in which to execute nodes on branches. Previously, n8n executed the first node of each branch, then the second of each branch, and so on (breadth-first). The new execution order ensures that each branch executes completely before starting the next one (depth-first). Branches execute based on their position on the canvas, from top to bottom. If two branches are at the same height, the leftmost one executes first.

n8n used to execute multi-input nodes as long as they received data on their first input. Nodes connected to the second input of multi-input nodes automatically executed regardless of whether they received data. The new execution order introduced in n8n 1.0

simplifies this behavior: Nodes are now executed only when they receive data, and multi-input nodes require data on at least one of their inputs to execute.

Your existing workflows will use the legacy order, while new workflows will execute using the v1 order. You can configure the execution order for each workflow in [workflow settings](#).

[PR #4238](#), [PR #6246](#), [PR #6507](#)

## Deprecations

### MySQL and MariaDB

n8n has deprecated support for MySQL and MariaDB as storage backends for n8n. These database systems are used by only a few users, yet they require continuous development and maintenance efforts. n8n recommends migrating to PostgreSQL for better compatibility and long-term support.

[PR #6189](#)

### `EXECUTIONS_PROCESS` and “own” mode

Previously, you could use the `EXECUTIONS_PROCESS` environment variable to specify whether executions should run in the `main` process or in their `own` processes. This option and `own` mode are now deprecated and will be removed in a future version of n8n. This is because it led to increased code complexity while offering marginal benefits. Starting from n8n 1.0, `main` will be the new default.

Note that executions start much faster in `main` mode than in `own` mode. However, if a workflow consumes more memory than is available, it might crash the entire n8n application instead of just the worker thread. To mitigate this, make sure to allocate enough system resources or configure [queue mode](#) to distribute executions among multiple workers.

[PR #6196](#)

## Breaking changes

### Docker

#### Permissions change

When using Docker-based deployments, the n8n process is now run by the user `node` instead of `root`. This change increases security.

If permission errors appear in your n8n container logs when starting n8n, you may need to update the permissions by executing the following command on the Docker host:

```
docker run --rm -it --user root -v ~/n8n:/home/node/.n8n --entrypoint chown n8nio/base:16 -R node:node /home/node/.n8n
```

#### Image removal

We've removed the Debian and RHEL images. If you were using these you need to change the image you use. This shouldn't result in any errors unless you were making a custom image based on one of those images.

### **Entrypoint change**

The entrypoint for the container has changed and you no longer need to specify the n8n command. If you were previously running `n8n worker --concurrency=5` it's now `worker --concurrency=5`

[PR #6365](#)

### **Workflow failures due to expression errors**

Workflow executions may fail due to syntax or runtime errors in expressions, such as those that reference non-existent nodes. While expressions already throw errors on the frontend, this change ensures that n8n also throws errors on the backend, where they were previously silently ignored. To receive notifications of failing workflows, n8n recommends setting up an "error workflow" under workflow settings.

[PR #6352](#)

### **Mandatory owner account**

This change makes [User Management](#) mandatory and removes support for other authentication methods, such as BasicAuth and External JWT. Note that the number of permitted users on [n8n.cloud](#) or custom plans still varies depending on your subscription.

[PR #6362](#)

### **Directory for installing custom nodes**

n8n will no longer load custom nodes from its global `node_modules` directory. Instead, you must install (or link) them to `~/.n8n/custom` (or a directory defined by `N8N_CUSTOM_EXTENSIONS`). Custom nodes that are npm packages will be located in `~/.n8n/nodes`. If you have custom nodes that were linked using `npm link` into the global `node_modules` directory, you need to link them again, into `~/.n8n/nodes` instead.

[PR #6396](#)

### **WebSockets**

The `N8N_PUSH_BACKEND` environment variable can be used to configure one of two available methods for pushing updates to the user interface: `sse` and `websocket`. Starting with n8n 1.0, `websocket` is the default method.

[PR #6196](#)

### **Date transformation functions**

n8n provides various transformation functions that operate on dates. These functions may return either a JavaScript Date or a Luxon DateTime object. With the new behavior, the return type always matches the input. If you call a date transformation function on a Date, it returns a Date. Similarly, if you call it on a DateTime object, it returns a DateTime object.

To identify any workflows and nodes that might be impacted by this change, you can use this [utility workflow](#).

For more information about date transformation functions, please refer to the [official documentation](#).

[PR #6435](#)

## Execution data retention

Starting from n8n 1.0, all successful, failed, and manual workflow executions will be saved by default. These settings can be modified for each workflow under “Workflow Settings,” or globally using the respective environment variables. Additionally, the EXECUTIONS\_DATA\_PRUNE setting will be enabled by default, with EXECUTIONS\_DATA\_PRUNE\_MAX\_COUNT set to 10,000. These default settings are designed to prevent performance degradation when using SQLite. Make sure to configure them according to your individual requirements and system capacity.

[PR #6577](#)

## Removed N8N\_USE\_DEPRECATED\_REQUEST\_LIB

The legacy request library has been deprecated for some time now. As of n8n 1.0, the ability to fall back to it in the HTTP Request node by setting the N8N\_USE\_DEPRECATED\_REQUEST\_LIB environment variable has been fully removed. The HTTP Request node will now always use the new HttpRequest interface.

If you build custom nodes, refer to [HTTP request helpers](#) for more information on migrating to the new interface.

[PR #6413](#)

## Removed WEBHOOK\_TUNNEL\_URL

As of version 0.227.0, n8n has renamed the WEBHOOK\_TUNNEL\_URL configuration option to WEBHOOK\_URL. In n8n 1.0, WEBHOOK\_TUNNEL\_URL has been removed. Update your setup to reflect the new name. For more information about this configuration option, refer to [the docs](#).

[PR #1408](#)

## Remove Node 16 support

n8n now requires Node 18.17.0 or above.

# Updating to n8n 1.0

1. Create a full backup of n8n.

2. n8n recommends updating to the latest n8n 0.x release before updating to n8n 1.x. This will allow you to pinpoint any potential issues to the correct release. Once you have verified that n8n 0.x starts up without any issues, proceed to the next step.
3. Carefully read the [Deprecations](#) and [Breaking Changes](#) sections above to assess how they may affect your setup.
4. Update to n8n 1.0:
  - During beta (before July 24th 2023): If using Docker, pull the next Docker image.
  - After July 24th 2023: If using Docker, pull the latest Docker image.
5. If you encounter any issues, redeploy the previous n8n version and restore the backup.

## Reporting issues

If you encounter any issues during the process of updating to n8n 1.0, please seek help in the community [forum](#).

## Thank you

We would like to take a moment to express our gratitude to all of our users for their continued support and feedback. Your contributions are invaluable in helping us make n8n the best possible automation tool. We're excited to continue working with you as we move forward with the release of version 1.0 and beyond. Thank you for being a part of our journey!

---

## Get help with n8n

### Where to get help

n8n provides different support options depending on your plan and the nature of your problem.

#### n8n community forum

n8n provides free community support for all n8n users through the [forum](#).

This is the best source for answers of all kinds, as both the n8n support team and community members can help.

#### Email support

n8n offers email support through the [help@n8n.io](mailto:help@n8n.io) for the following plans:

- [Enterprise plans](#) can use email support with an SLA for technical, account, billing, and other inquiries.
- Other [Cloud plans](#) can use email support for admin and billing issues. For technical support, please refer to the forum.

## What to include in your message

When posting to the forum or emailing customer support, you'll get help faster if you provide details in your first message about your n8n instance and the issue you're experiencing.

### Your n8n instance details

To collect basic information about your n8n instance:

1. Open the left-side panel.
2. Select **Help**.
3. Select **About n8n**.
4. The **About n8n** modal opens to display your current information.
5. Select **Copy debug information** to copy your information.
6. Include this information in your forum post or support email.

### Details about your problem

To help resolve your issues more efficiently, here are some things you can include to provide more context:

- :video\_camera: **Screenshots or video recordings**: A quick Loom or screen recording that shows what's happening.
- :books: **Relevant documentation**: If you've followed any guides or documentation, include links to them in your message.
- :cloud: **n8n Cloud workspace (if possible)**: If contacting support, provide the workspace URL for your n8n Cloud instance. It looks something like <https://xxxxx.n8n.app.cloud>.
- :memo: **Steps to reproduce the issue**: A simple step-by-step outline of what you did before encountering the issue.
- :open\_file\_folder: **Workflow or Configuration files**: Sharing relevant workflows or configuration files can be a huge help.

It may also be helpful to include a [HAR \(HTTP Archive\) file](#) in your message. You can learn how to generate a HAR file in your browser and how to redact sensitive details before posting using the [Har Analyzer](#).

---

## How can you contribute?

There are several ways in which you can contribute to n8n, depending on your skills and interests. Each form of contribution is valuable to us!

### Share some love: Review us

- Star n8n on [GitHub](#) and [Docker Hub](#).
- Follow us on [Twitter](#), [LinkedIn](#), and [Facebook](#).
- Upvote n8n on [AlternativeTo](#) and [Alternative.me](#).
- Add n8n to your stack on [Stackshare](#).
- Write a review about n8n on [G2](#), [Slant](#), and [Capterra](#).

## Help out the community

You can participate in the [forum](#) and help the community members out with their questions.

When sharing workflows in the community forum for debugging, use code blocks. Use triple backticks ` ``` to wrap the workflow JSON in a code block.

The following video demonstrates the steps of sharing workflows on the community forum:

## Contribute a workflow template

-8<- “ \_snippets/workflows/templates/submit-templates.md”

## Build a node

Create an integration for a third party service. Check out [the node creation docs](#) for guidance on how to create and publish a community node.

## Contribute to the code

There are different ways in which you can contribute to the n8n code base:

- Fix [issues](#) reported on GitHub. The [CONTRIBUTING](#) guide will help you get your development environment ready in minutes.
- Add additional functionality to an existing third party integration.
- Add a new feature to n8n.

## Contribute to the docs

You can contribute to the n8n documentation, for example by documenting nodes or fixing issues.

The repository for the docs is [here](#) and the guidelines for contributing to the docs are [here](#).

## Contribute to community tutorials

Share your own video or written guides on our [community-driven, searchable library of n8n tutorials and training materials](#). Tag them for easy discovery, and post in your language's subcategory. Follow the contribution guidelines to help keep our growing library high-quality and accessible to everyone.

## Refer a candidate

Do you know someone who would be a great fit for one of our open positions? Refer them to us! In return, we'll pay you €1,000 when the referral successfully passes their probationary period.

Here's how this works:

1. **Search:** Have a look at the description and requirements of each role, and consider if someone you know would be a great fit.
  2. **Referral:** Once you've identified a potential candidate, send an email to [Jobs at n8n](#) with the subject line *Employee referral - [job title]* and a short description of the person you're referring (and the reason why). Also, tell your referral to apply for the job through our [careers page](#).
  3. **Evaluation:** We'll screen the application and inform you about the next steps of the hiring process.
  4. **Reward:** As soon as your referral has successfully finished the probationary period, we'll reward you for your efforts by transferring the €1,000 to your bank account.
- 

## Privacy and security at n8n

n8n is committed to the privacy and security of your data. This section outlines how n8n handles and secures data. This isn't an exhaustive list of practices, but an overview of key policies and procedures.

If you have any questions related to data privacy, email [privacy@n8n.io](mailto:privacy@n8n.io).

If you have any security-related questions, or if you want to report a suspected vulnerability, email [security@n8n.io](mailto:security@n8n.io).

---

## Privacy

This page describes n8n's data privacy practices.

### GDPR

#### Data processing agreement

For Cloud versions of n8n, n8n is considered both a Controller and a Processor as defined by the GDPR. As a Processor, n8n implements policies and practices that secure the personal data you send to the platform, and includes a [Data Processing Agreement](#) as part of the company's standard [Terms of Service](#).

The n8n Data Processing Agreement includes the [Standard Contractual Clauses \(SCCs\)](#). These clarify how n8n handles your data, and they update n8n's GDPR policies to cover the latest standards set by the European Commission.

You can find a list of n8n sub-processors [here](#).

#### Submitting an account deletion request

Email [help@n8n.io](mailto:help@n8n.io) to make an account deletion request.

## Sub-processors

This is a list of sub-processors authorized to process customer data for n8n's service. n8n audits each sub-processor's security controls and applicable regulations for the protection of personal data.

<b>Sub-processor name</b>	<b>Purpose</b>	<b>Contact details</b>
Microsoft Azure	Cloud service provider	Microsoft Azure 1 Microsoft Way Redmond WA 98052 USA Contact information: <a href="https://privacy.microsoft.com/en-GB/privacystatement#mainhowtocontactusmc">https://privacy.microsoft.com/en-GB/privacystatement#mainhowtocontactusmc</a>
Hetzner Online	Cloud service provider	Hetzner Online GmbH Industriestr. 25 91710 Gunzenhausen Germany <a href="mailto:data-protection@hetzner.com">data-protection@hetzner.com</a>
OpenAI	AI provider	1455 3rd Street San Francisco, CA 94158 United States
Anthropic	AI provider	Anthropic Ireland, Limited 6th Floor South Bank House, Barrow Street, Dublin 4 Ireland
Google Vertex AI	AI provider	Google LLC, 1600 Amphitheatre Parkway, Mountain View, CA 94043, United States
LangChain	AI provider	LangChain, Inc. Delaware

Subscribe [here](#) to receive updates when n8n adds or changes a sub-processor.

## GDPR for self-hosted users

-8<- “\_snippets/privacy-security/gdpr-self-hosted.md”

## Data collection

n8n collects selected usage and performance data to help diagnose problems and improve the platform. Read about how n8n stores and processes this information in the [privacy policy](#).

The data gathered is different in self-hosted n8n and n8n Cloud.

## Data collection in self-hosted n8n

n8n takes care to keep self-hosted data anonymous and avoids collecting sensitive data.

## What n8n collects

- Error codes and messages of failed executions (excluding any payload data, and not for custom nodes)
- Error reports for app crashes and API issues
- The graph of a workflow (types of nodes used and how they're connected)
- From node parameters:
  - The 'resource' and 'operation' that a node is set to (if applicable)
  - For HTTP request nodes, the domain, path, and method (with personal data anonymized)
- Data around workflow executions:
  - Status
  - The user ID of the user who ran the execution
  - The first time a workflow loads data from an external source
  - The first successful production (non-manual) workflow execution
- The domain of webhook calls, if specified (excluding subdomain).
- Details on how the UI is used (for example, navigation, nodes panel searches)
- Diagnostic information:
  - n8n version
  - Selected settings:
    - DB\_TYPE
    - N8N\_VERSION\_NOTIFICATIONS\_ENABLED
    - N8N\_DISABLE\_PRODUCTION\_MAIN\_PROCESS
    - Execution variables
  - OS, RAM, and CPUs
  - Anonymous instance ID
- IP address

## What n8n doesn't collect

n8n doesn't collect private or sensitive information, such as:

- Personally identifiable information (except IP address)
- Credential information
- Node parameters (except 'resource' and 'operation')
- Execution data
- Sensitive settings (for example, endpoints, ports, DB connections, username/password)
- Error payloads

## How collection works

Most data is sent to n8n as events that generate it occur. Workflow execution counts and an instance pulse are sent periodically (every 6 hours).

## Opting out of telemetry

Telemetry collection is enabled by default. To disable it you can configure the following environment variables.

To opt out of telemetry events:

```
export N8N_DIAGNOSTICS_ENABLED=false
```

To opt out of checking for new versions of n8n:

```
export N8N_VERSION_NOTIFICATIONS_ENABLED=false
```

To disable the templates feature (prevents background health check calls):

```
export N8N_TEMPLATES_ENABLED=false
```

See [configuration](#) for more info on how to set environment variables.

## Data collection in n8n Cloud

n8n Cloud collects everything listed in [Data collection in self-hosted n8n](#).

Additionally, in n8n Cloud, n8n uses [PostHog](#) to track events and visualise usage, including using session recordings. Session recordings comprise the data seen by a user on screen, with the exception of credential values. n8n's product team uses this data to improve the product. All recordings are deleted after 21 days.

## AI in n8n

To provide enhanced assistance, n8n integrates AI-powered features that leverage Large Language Models (LLMs).

### How n8n uses AI

To assist and improve user experience, n8n may send specific context data to LLMs. This context data is strictly limited to information about the current workflow. n8n does not send any values from credential fields or actual output data to AI services. The data will not be incorporated, used, or retained to train the models of the AI services. Any data will be deleted after 30 days.

### When n8n shares data

Data is only sent to AI services if workspaces have opted in to use the assistant. The Assistant is enabled by default for n8n Cloud users. When a workspace opts in to use the assistant, node-specific data is transmitted only during direct interactions and active sessions with the AI assistant, ensuring no unnecessary data sharing occurs.

### What n8n shares

- **General Workflow Information:** This includes details about which nodes are present in your workflow, the number of items currently in the workflow, and whether the workflow is active.
- **Input & Output Schemas of Nodes:** This includes the schema of all nodes with incoming data and the output schema of a node in question. We do not send the actual data value of the schema.
- **Node Configuration:** This includes the operations, options, and settings chosen in the referenced node.
- **Code and Expressions:** This includes any code or expressions in the node in question to help with debugging potential issues and optimizations.

## **What n8n doesn't share**

- **Credentials:** Any values of the credential fields of your nodes.
- **Output Data:** The actual data processed by your workflows.
- **Sensitive Information:** Any personally identifiable information or other sensitive data that could compromise your privacy or security that you have not explicitly mentioned in node parameters or your code of a [Code Node](#).

## **Documentation telemetry**

n8n's documentation (this website) uses cookies to recognize your repeated visits and preferences, as well as to measure the effectiveness of n8n's documentation and whether users find what they're searching for. With your consent, you're helping n8n to make our documentation better. You can control cookie consent using the cookie widget.

## **Retention and deletion of personal identifiable data**

PID (personal identifiable data) is data that's personal to you and would identify you as an individual.

### **n8n Cloud**

#### **PID retention**

n8n only retains data for as long as necessary to provide the core service.

For n8n Cloud, n8n stores your workflow code, credentials, and other data indefinitely, until you choose to delete it or close your account. The platform stores execution data according to the retention rules on your account.

n8n deletes most internal application logs and logs tied to subprocessors within 90 days. The company retains a subset of logs for longer periods where required for security investigations.

#### **PID deletion**

If you choose to delete your n8n account, n8n deletes all customer data and event data associated with your account. n8n deletes customer data in backups within 90 days.

### **Self-hosted**

Self-hosted users should have their own PID policy and data deletion processes. Refer to [What you can do](#) for more information.

## **Payment processor**

n8n uses Paddle.com to process payments. When you sign up for a paid plan, Paddle transmits and stores the details of your payment method according to their security policy. n8n stores no information about your payment method.

---

## Incident response

n8n implements incident response best practices for identifying, documenting, resolving and communicating incidents.

n8n publishes incident notifications to a status page at [n8n Status](#).

n8n notifies customers of any data breaches according to the company's [Data Processing Addendum](#).

---

## What you can do

It's also your responsibility as a customer to ensure you are securing your code and data. This document lists some steps you can take.

### All users

- Report security issues and [terms of service](#) violations to security@n8n.io.
- If more than one person uses your n8n instance, set up [User management](#) and follow the [Best practices](#).
- Use OAuth to connect integrations whenever possible.

### Self-hosted users

If you self-host n8n, there are additional steps you can take:

- Set up a reverse proxy to handle TLS, ensuring data is encrypted in transit.
- Ensure data is encrypted at rest by using encrypted partitions, or encryption at the hardware level, and ensuring n8n and its database is written to that location.
- Run a [Security audit](#).
- Be aware of the [Risks](#) when installing community nodes, or choose to disable them.
- Make sure users can't import external modules in the Code node. Refer to [Environment variables | Nodes](#) for more information.
- Choose to exclude certain nodes. For example, you can disable nodes like Execute Command or SSH. Refer to [Environment variables | Nodes](#) for more information.
- For maximum privacy, you can [Isolate n8n](#).

### GDPR for self-hosted users

-8<- “\_snippets/privacy-security/gdpr-self-hosted.md”

---

# Sustainable Use License

n8n's [Sustainable Use License](#) and [n8n Enterprise License](#) are based on the [fair-code](#) model.

## License FAQs

### What license do you use?

n8n uses the [Sustainable Use License](#) and [n8n Enterprise License](#). These licenses are based on the [fair-code](#) model.

### What source code is covered by the Sustainable Use License?

The [Sustainable Use License](#) applies to all our source code hosted in our [main GitHub repository](#) except:

- Content of branches other than master.
- Source code files that contain .ee. in their file name. These are licensed under the [n8n Enterprise License](#).

### What is the Sustainable Use License?

The Sustainable Use License is a fair-code software license created by n8n in 2022. You can read more about why we did this [here](#). The license allows you the free right to use, modify, create derivative works, and redistribute, with three limitations:

- You may use or modify the software only for your own internal business purposes or for non-commercial or personal use.
- You may distribute the software or provide it to others only if you do so free of charge for non-commercial purposes.
- You may not alter, remove, or obscure any licensing, copyright, or other notices of the licensor in the software. Any use of the licensor's trademarks is subject to applicable law.

We encourage anyone who wants to use the Sustainable Use License. If you are building something out in the open, it makes sense to think about licensing earlier in order to avoid problems later. Contact us at [license@n8n.io](mailto:license@n8n.io) if you would like to ask any questions about it.

### What is and isn't allowed under the license in the context of n8n's product?

Our license restricts use to "internal business purposes". In practice this means all use is allowed unless you are selling a product, service, or module in which the value derives entirely or substantially from n8n functionality. Here are some examples that wouldn't be allowed:

- White-labeling n8n and offering it to your customers for money.
- Hosting n8n and charging people money to access it.

All of the following examples are allowed under our license:

- Using n8n to sync the data you control as a company, for example

- from a CRM to an internal database.
- Creating an n8n node for your product or any other integration between your product and n8n.
  - Providing consulting services related to n8n, for example building workflows, custom features closely connect to n8n, or code that gets executed by n8n.
  - Supporting n8n, for example by setting it up or maintaining it on an internal company server.

### **Can I use n8n to act as the back-end to power a feature in my app?**

Usually yes, as long as the back-end process doesn't use users' own credentials to access their data.

Here are two examples to clarify:

#### ***Example 1: Sync ACME app with HubSpot***

Bob sets up n8n to collect a user's HubSpot credentials to sync data in the ACME app with data in HubSpot.

**NOT ALLOWED** under the Sustainable Use License. This use case collects the user's own HubSpot credentials to pull information to feed into the ACME app.

#### ***Example 2: Embed AI chatbot in ACME app***

Bob sets up n8n to embed an AI chatbot within the ACME app. The AI chatbot's credentials in n8n use Bob's company credentials. ACME app end-users only enter their questions or queries to the chatbot.

**ALLOWED** under the Sustainable Use License. No user credentials are being collected.

### **What if I want to use n8n for something that's not permitted by the license?**

You must sign a separate commercial agreement with us. We actively encourage software creators to embed n8n within their products; we just ask them to sign an agreement laying out the terms of use, and the fees owed to n8n for using the product in this way. We call this mode of use n8n Embed. You can learn more, and contact us about it [here](#).

If you are unsure whether the use case you have in mind constitutes an internal business purpose or not, take a look at [the examples](#), and if you're still unclear, email us at [license@n8n.io](mailto:license@n8n.io).

### **Why don't you use an open source license?**

n8n's mission is to give everyone who uses a computer technical superpowers. We've decided the best way for us to achieve this mission is to make n8n as widely and freely available as possible for users, while ensuring we can build a sustainable, viable business. By making our product free to use, easy to distribute, and source-

available we help everyone access the product. By operating as a business, we can continue to release features, fix bugs, and provide reliable software at scale long-term.

## **Why did you create a license?**

Creating a license was our least favorite option. We only went down this path after reviewing the possible existing licenses and deciding nothing fit our specific needs. There are two ways in which we try to mitigate the pain and friction of using a proprietary license:

1. By using plain English, and keeping it as short as possible.
2. By promoting [fair-code](#) with the goal of making it a well-known umbrella term to describe software models like ours.

Our goals when we created the Sustainable Use License were:

1. To be as permissive as possible.
2. Safeguarding our ability to build a business.
3. Being as clear as possible what use was permitted or not.

## **My company has a policy against using code that restricts commercial use - can I still use n8n?**

Provided you are using n8n for internal business purposes, and not making n8n available to your customers for them to connect their accounts and build workflows, you should be able to use n8n. If you are unsure whether the use case you have in mind constitutes an internal business purpose or not, take a look at [the examples](#), and if you're still unclear, email us at [license@n8n.io](mailto:license@n8n.io).

## **What happens to the code I contribute to n8n in light of the Sustainable Use License?**

Any code you contribute on GitHub is subject to GitHub's [terms of use](#). In simple terms, this means you own, and are responsible for, anything you contribute, but that you grant other GitHub users certain rights to use this code. When you contribute code to a repository containing notice of a license, you license the code under the same terms.

n8n asks every contributor to sign our [Contributor License Agreement](#). In addition to the above, this gives n8n the ability to change its license without seeking additional permission. It also means you aren't liable for your contributions (e.g. in case they cause damage to someone else's business).

It's easy to get started contributing code to n8n [here](#), and we've listed broader ways of participating in our community [here](#).

## **Why did you switch to the Sustainable Use License from your previous license arrangement (Apache 2.0 with Commons Clause)?**

n8n was licensed under Apache 2.0 with Commons Clause until 17 March 2022. Commons Clause was initiated by various software companies wanting to protect their rights against cloud providers. The

concept involved adding a commercial restriction on top of an existing open source license.

However, the use of the Commons Clause as an additional condition to an open source license, as well as the use of wording that's open to interpretation, created some confusion and uncertainty regarding the terms of use. The Commons Clause also restricted people's ability to offer consulting and support services: we realized these services are critical in enabling people to get value from n8n, so we wanted to remove this restriction.

We created the Sustainable Use License to be more permissive and more clear about what use is allowed, while continuing to ensure n8n gets the funding needed to build and improve our product.

### **What are the main differences between the Sustainable Use License and your previous license arrangement (Apache 2.0 with Commons Clause)?**

There are two main differences between the Sustainable Use License and our previous license arrangement. The first is that we have tightened the definition of how you can use the software. Previously the Commons Clause restricted users ability to "sell" the software; we have redefined this to restrict use to internal business purposes. The second difference is that our previous license restricted people's ability to charge fees for consulting or support services related to the software: we have lifted that restriction altogether.

That means you are now free to offer commercial consulting or support services (e.g. building n8n workflows) without the need for a separate license agreement with us. If you are interested in joining our community of n8n experts providing these services, you can learn more [here](#).

### **Is n8n open source?**

Although n8n's source code is available under the Sustainable Use License, according to the [Open Source Initiative](#) (OSI), open source licenses can't include limitations on use, so we do not call ourselves open source. In practice, n8n offers most users many of the same benefits as OSI-approved open source.

We coined the term '[fair-code](#)' as a way of describing our licensing model, and the model of other companies who are source-available, but restrict commercial use of their source code.

### **What is fair-code, and how does the Sustainable Use License relate to it?**

Fair-code isn't a software license. It describes a software model where software:

- Is generally free to use and can be distributed by anybody.
- Has its source code openly available.
- Can be extended by anybody in public and private communities.
- Is commercially restricted by its authors.

The Sustainable Use License is a fair-code license. You can read more about it and see other examples of fair-code licenses [here](#).

We're always excited to talk about software licenses, fair-code, and other principles around sharing code with interested parties. To get in touch to chat, email [license@n8n.io](mailto:license@n8n.io).

## Can I use n8n's Sustainable Use License for my own project?

Yes! We're excited to see more software use the Sustainable Use License. We'd love to hear about your project if you're using our license: [license@n8n.io](mailto:license@n8n.io).

---

# Integrations

n8n calls integrations nodes.

Nodes are the building blocks of workflows in n8n. They're an entry point for retrieving data, a function to process data, or an exit for sending data. The data process includes filtering, recomposing, and changing data. There can be one or several nodes for your API, service or app. You can connect multiple nodes, which allows you to create complex workflows.

## Built-in nodes

n8n includes a collection of built-in integrations. Refer to [Built-in nodes](#) for documentation on all n8n's built-in nodes.

## Community nodes

As well as using the built-in nodes, you can also install community-built nodes. Refer to [Community nodes](#) for more information.

## Credential-only nodes and custom operations

-8<- “\_snippets/integrations/credential-only-intro.md”

Refer to [Custom operations](#) for more information.

## Generic integrations

If you need to connect to a service where n8n doesn't have a node, or a credential-only node, you can still use the [HTTP Request](#) node. Refer to the node page for details on how to set up authentication and create your API call.

## Where to go next

- If you want to create your own node, head over to the [Creating Nodes](#) section.

- Check out [Community nodes](#) to learn about installing and managing community-built nodes.
  - If you'd like to learn more about the different nodes in n8n, their functionalities and example usage, check out n8n's node libraries: [Core nodes](#), [Actions](#), and [Triggers](#).
  - If you'd like to learn how to add the credentials for the different nodes, head over to the [Credentials](#) section.
- 

## Built-in integrations

This section contains the [node](#) library: reference documentation for every built-in node in n8n, and their credentials.

-8<- “\_snippets/integrations/builtin/node-operations.md”

### Core nodes

Core nodes can be actions or [triggers](#). Whereas most nodes connect to a specific external service, core nodes provide functionality such as logic, scheduling, or generic API calls.

### Cluster nodes

-8<- “\_snippets/integrations/builtin/cluster-nodes/cluster-nodes-summary.md”

### Credentials

External services need a way to identify and authenticate users. This data can range from an API key over an email/password combination to a long multi-line private key. You can save these in n8n as [credentials](#).

Nodes in n8n can then request that credential information. As another layer of security, only node types with specific access rights can access the credentials.

To make sure that the data is secure, it gets saved to the database encrypted. n8n uses a random personal encryption key, which it automatically generates on the first run of n8n and then saved under `~/.n8n/config`.

To learn more about creating, managing, and sharing credentials, refer to [Manage credentials](#).

### Community nodes

n8n supports custom nodes built by the community. Refer to [Community nodes](#) for guidance on installing and using these nodes.

For help building your own custom nodes, and publish them to [npm](#), refer to [Creating nodes](#) for more information.

---

# Core nodes library

This section provides information about n8n's core [nodes](#).

---

## Activation Trigger node

The Activation Trigger node gets triggered when an event gets fired by n8n or a workflow.

The Activation Trigger node gets triggered for the workflow that it gets added to. You can use the Activation Trigger node to trigger a workflow to notify the state of the workflow.

### Node parameters

- Events
  - **Activation**: Run when the workflow gets published
  - **Start**: Run when n8n starts or restarts
  - **Update**: Run when the workflow gets saved while it's active

## Templates and examples

---

## Aggregate

Use the Aggregate node to take separate items, or portions of them, and group them together into individual items.

### Node parameters

To begin using the node, select the **Aggregate** you'd like to use:

- Individual Fields: Aggregate individual fields separately.
- All Item Data: Aggregate all item data into a single list.

### Individual Fields

- **Input Field Name**: Enter the name of the field in the input data to aggregate together.
- **Rename Field**: This toggle controls whether to give the field a different name in the aggregated output data. Turn this on to add a different field name. If you're aggregating multiple fields, you must provide new output field names. You can't leave multiple fields undefined.
  - **Output Field Name**: This field is displayed when you turn on **Rename Field**. Enter the field name for the aggregated output data.

Refer to [Node options](#) for more configuration options.

## All Item Data

- **Put Output in Field:** Enter the name of the field to output the data in.
- **Include:** Select which fields to include in the output. Choose from:
  - **All fields:** The output includes data from all fields with no further parameters.
  - **Specified Fields:** If you select this option, enter a comma-separated list of fields the output should include data from in the **Fields To Include** parameter. The output will include only the fields in this list.
  - **All Fields Except:** If you select this option, enter a comma-separated list of fields the output should exclude data from in the **Fields To Exclude** parameter. The output will include all fields not in this list.

Refer to [Node options](#) for more configuration options.

## Node options

You can further configure this node using these **Options**:

- **Disable Dot Notation:** The node displays this toggle when you select the **Individual Fields** Aggregate. It controls whether to disallow referencing child fields using parent.child in the field name (turned on), or allow it (turned off, default).
- **Merge Lists:** The node displays this toggle when you select the **Individual Fields** Aggregate. Turn it on if the field to aggregate is a list and you want to output a single flat list rather than a list of lists.
- **Include Binaries:** The node displays this toggle for both Aggregate types. Turn it on if you want to include binary data from the input in the new output.
- **Keep Missing And Null Values:** The node displays this toggle when you select the **Individual Fields** Aggregate. Turn it on to add a null (empty) entry in the output list when there is a null or missing value in the input. If turned off, the output ignores null or empty values.

## Templates and examples

## Related resources

-8<- “\_snippets/integrations/builtin/core-nodes/data-transformation-actions/data-section-link.md”

---

## AI Transform

Use the AI Transform node to generate code snippets based on your prompt. The AI is context-aware, understanding the workflow's nodes and their data types.

## Node parameters

### Instructions

Enter your prompt for the AI and click the **Generate code** button to automatically populate the **Transformation Code**. For example, you can specify how you want to process or categorize your data. Refer to [Writing good prompts](#) for more information.

The prompt should be in plain English and under 500 characters.

### Transformation Code

The code snippet generated by the node is read-only. To edit this code, adjust your prompt in **Instructions** or copy and paste it into a [Code](#) node.

## Templates and examples

---

## Code node

```
-8<- "_snippets/integrations/builtin/core-nodes/code-node.md"
```

### Common issues

For common questions or issues and suggested solutions, refer to [Common Issues](#).

---

## Keyboard shortcuts when using the Code editor

The Code node editing environment supports a range of keyboard shortcuts to speed up and enhance your experience. Select the appropriate tab to see the relevant shortcuts for your operating system.

## Cursor Movement

==== "Windows"

Action	Shortcut
Move cursor left	++left++
Move cursor right	++right++
Move cursor up	++up++
Move cursor down	++down++
Move cursor by word left	++control+left++
Move cursor by word right	++control+right++

Move to line start	++home++ **or** ++control+left++
Move to line end	++end++ or ++control+right++
Move to document start	++control+home++
Move to document end	++control+end++
Move page up	++page-up++
Move page down	++page-down++

==== "macOS"

Action	Shortcut
-	-
Move cursor left	++left++ **or** ++control+b++
Move cursor right	++right++ **or** ++control+f++
Move cursor up	++up++ **or** ++control+p++
Move cursor down	++down++ **or** ++control+n++
Move cursor by word left	++option+left++
Move cursor by word right	++option+right++
Move to line start	++command+left++ **or** ++control+a++
Move to line end	++command+right++ **or** ++control+e++
Move to document start	++command+up++
Move to document end	++command+down++
Move page up	++page-up++ **or** ++option+v++
Move page down	++page-down++ **or** ++control+v++

==== "Linux"

Action	Shortcut
-	-
Move cursor left	++left++
Move cursor right	++right++
Move cursor up	++up++
Move cursor down	++down++
Move cursor by word left	++control+left++
Move cursor by word right	++control+right++
Move to line start	++home++ **or** ++control+left++
Move to line end	++end++ or ++control+right++
Move to document start	++control+home++
Move to document end	++control+end++
Move page up	++page-up++
Move page down	++page-down++

## Selection

==== "Windows"

Action	Shortcut
-	-

Selection with any movement key	++shift++ + [Movement Key]
Select all	++control+a++
Select line	++control+l++
Select next occurrence	++control+d++
Select all occurrences	++shift+control+l++
Go to matching bracket	++shift+control+backslash++

==== "macOS"

Action	Shortcut
Selection with any movement key	++shift++ + [Movement Key]
Select all	++command+a++
Select line	++command+l++
Select next occurrence	++command+d++
Go to matching bracket	++shift+command+backslash++

==== "Linux"

Action	Shortcut
Selection with any movement key	++shift++ + [Movement Key]
Select all	++control+a++
Select line	++control+l++
Select next occurrence	++control+d++
Select all occurrences	++shift+control+l++
Go to matching bracket	++shift+control+backslash++

## Basic Operations

==== "Windows"

Action	Shortcut
- - -	
New line with indentation	++enter++
Undo	++control+z++
Redo	++control+y++ **or**
++control+shift+z++	
Undo selection	++control+u++
Copy	++control+c++
Cut	++control+x++
Paste	++control+v++

==== "macOS"

Action	Shortcut
- - -	
New line with indentation	++enter++

Undo	++command+z++
Redo	++command+y++ **or**
++command+shift+z++	
Undo selection	++command+u++
Copy	++command+c++
Cut	++command+x++
Paste	++command+v++

==== "Linux"

Action	Shortcut
----- -----	
--	
New line with indentation	++enter++
Undo	++control+z++
Redo	++control+y++ **or**
++control+shift+z++	
Undo selection	++control+u++
Copy	++control+c++
Cut	++control+x++
Paste	++control+v++

## Delete Operations

==== "Windows"

Action	Shortcut	
----- -----		
Delete character left	++backspace++	
Delete character right	++delete++	
Delete word left	++control+backspace++	
Delete word right	++control+delete++	
Delete line	++shift+control+k++	

==== "macOS"

Action	Shortcut	
----- -----		
--		
Delete character left	++backspace++	
Delete character right	++delete++	
Delete word left	++option+backspace++ **or**	
++control+command+h++		
Delete word right	++option+delete++ **or**	
++function+option+backspace++		

Delete line	++shift+command+k++
Delete to line start	++command+backspace++
Delete to line end	++command+delete++ **or** ++control+k++

==== "Linux"

Action	Shortcut
Delete character left	++backspace++
Delete character right	++delete++
Delete word left	++control+backspace++
Delete word right	++control+delete++
Delete line	++shift+control+k++

## Line Operations

==== "Windows"

Action	Shortcut
Move line up	++alt+up++
Move line down	++alt+down++
Copy line up	++shift+alt+up++
Copy line down	++shift+alt+down++
Toggle line comment	++control+slash++
Add line comment	++control+k++ **then** ++control+c++
Remove line comment	++control+k++ **then** ++control+u++
Toggle block comment	++shift+alt+a++

==== "macOS"

Action	Shortcut
Move line up	++option+up++
Move line down	++option+down++
Copy line up	++shift+option+up++
Copy line down	++shift+option+down++
Toggle line comment	++command+slash++
Add line comment	++command+k++ **then** ++command+c++
Remove line comment	++command+k++ **then** ++command+u++
Toggle block comment	++shift+option+a++
Split line	++control+o++
Transpose characters	++control+t++

==== "Linux"

Action	Shortcut
Move line up	++alt+up++
Move line down	++alt+down++
Copy line up	++shift+alt+up++
Copy line down	++shift+alt+down++
Toggle line comment	++control+slash++
Add line comment	++control+k++ **then** ++control+c++
Remove line comment	++control+k++ **then** ++control+c++
Toggle block comment	++shift+alt+a++

## Autocomplete

==== "Windows"

Action	Shortcut
Start completion	++control+space++
Accept completion	++enter++ **or** ++tab++
Close completion	++escape++
Navigate completion options	++up++ **or** ++down++

==== "macOS"

Action	Shortcut
Start completion	++control+space++
Accept completion	++enter++ **or** ++tab++
Close completion	++escape++
Navigate completion options	++up++ **or** ++down++

==== "Linux"

Action	Shortcut
Start completion	++control+space++
Accept completion	++enter++ **or** ++tab++
Close completion	++escape++
Navigate completion options	++up++ **or** ++down++

## Indentation

==== "Windows"

Action	Shortcut
Indent more	++tab++ **or** ++control+bracket-right++
Indent less	++shift+tab++ **or** ++control+bracket-left++

==== "macOS"

Action	Shortcut
Indent more	++command+bracket-right++
Indent less	++command+bracket-left++

==== "Linux"

Action	Shortcut
Indent more	++tab++ **or** ++control+bracket-right++
Indent less	++shift+tab++ **or** ++control+bracket-left++

## Code Folding

==== "Windows"

Action	Shortcut
Fold code	++control+shift+bracket-left++

Unfold code	++control+shift+bracket-right++	
Fold all	++control+k++ **then** ++control+0++	
Unfold all	++control+k++ **then** ++control+j++	

==== "macOS"

Action	Shortcut	
----- -----		
Fold code	++command+option+bracket-left++	
Unfold code	++command+option+bracket-right++	
Fold all	++command+k++ **then** ++command+0++	
Unfold all	++command+k++ **then** ++command+j++	

==== "Linux"

Action	Shortcut	
----- -----		
Fold code	++control+shift+bracket-left++	
Unfold code	++control+shift+bracket-right++	
Fold all	++control+k++ **then** ++control+0++	
Unfold all	++control+k++ **then** ++control+j++	

## Multicursor

==== "Windows"

Action	Shortcut	
----- -----		
Add cursor at click position	++alt+left-button++	
Add cursor above	++control+alt+up++	
Add cursor below	++control+alt+down++	
Add cursors to line ends	++shift+alt+i++	
Clear multiple cursors	++escape++	

==== "macOS"

Action	Shortcut	
----- -----		
Add cursor at click position	++option+left-button++	
Add cursor above	++control+option+up++	
Add cursor below	++control+option+down++	
Add cursors to line ends	++shift+option+i++	
Clear multiple cursors	++escape++	

==== "Linux"

Action	Shortcut	
----- -----		
Add cursor at click position	++alt+left-button++	
Add cursor above	++shift+alt+up++	
Add cursor below	++shift+alt+down++	
Add cursors to line ends	++shift+alt+i++	
Clear multiple cursors	++escape++	

## Formatting

==== "Windows"

Action	Shortcut	
----- -----		

-----   -----		
Format document   ++shift+alt+f++		
<b>==== "macOS"</b>		
Action             Shortcut		
<hr/>		
Format document   ++shift+command+f++		
<b>==== "Linux"</b>		
Action             Shortcut		
<hr/>		
Format document   ++control+shift+i++		

## Search & Navigation

Action             Shortcut		
<hr/>		
Open Search       ++control+f++		
Select All        ++alt+enter++		
Replace All       ++control+alt+enter++		
Go To Line        ++control+g++		
Next Diagnostic   ++f8++		
Previous Diag.   ++shift+f8++		
Open Lint Panel   ++control+shift+m++		
<b>==== "macOS"</b>		
Action             Shortcut		
<hr/>		
Open Search       ++command+f++		
Select All        ++command+enter++		
Replace All       ++command+option+enter++		
Go To Line        ++command+g++		
Next Diagnostic   ++f8++		
Previous Diag.   ++shift+f8++		
Open Lint Panel   ++command+shift+m++		
<b>==== "Linux"</b>		
Action             Shortcut		
<hr/>		
Open Search       ++control+f++		
Select All        ++alt+enter++		
Replace All       ++control+alt+enter++		
Go To Line        ++control+g++		
Next Diagnostic   ++f8++		
Previous Diag.   ++shift+f8++		
Open Lint Panel   ++control+shift+m++		

---

## Code node common issues

Here are some common errors and issues with the [Code node](#) and steps to resolve or troubleshoot them.

## Code doesn't return items properly

This error occurs when the code in your Code node doesn't return data in the expected format.

In n8n, all data passed between nodes is an array of objects. Each of these objects wraps another object with the `json` key:

```
[
 {
 "json": {
 // your data goes here
 }
 }
]
```

To troubleshoot this error, check the following:

- Read the [data structure](#) to understand the data you receive in the Code node and the requirements for outputting data from the node.
- Understand how data items work and how to connect data items from previous nodes with [item linking](#).

## A 'json' property isn't an object

This error occurs when the Code node returns data where the `json` key isn't pointing to an object.

This may happen if you set `json` to a different data structure, like an array:

```
[
 {
 "json": [
 // Setting `json` to an array like this will produce an error
]
 }
]
```

To resolve this, ensure that the `json` key references an object in your return data:

```
[
 {
 "json": {
 // Setting `json` to an object as expected
 }
 }
]
```

## Code doesn't return an object

This error may occur when your Code node doesn't return anything or if it returns an unexpected result.

To resolve this, ensure that your Code node returns the [expected data structure](#):

```
[
 {
 "json": {
 // your data goes here
 }
 }
]
```

This error may also occur if the code you provided returns 'undefined' instead of the expected result. In that case, ensure that the data you are referencing in your Code node exists in each execution and that it has the structure your code expects.

## 'import' and 'export' may only appear at the top level

This error occurs if you try to use `import` or `export` in the Code node. These aren't supported by n8n's JavaScript sandbox. Instead, use the `require` function to load modules.

To resolve this issue, try changing your `import` statements to use `require`:

```
// Original code:
// import express from "express";
// New code:
const express = require("express");
```

## Cannot find module '<module>'

This error occurs if you try to use `require` in the Code node and n8n can't find the module.

If you're self-hosting n8n, follow these steps:

- Install the module into your n8n environment.
  - If you are running n8n with npm, install the module in the same environment as n8n.
  - If you are running n8n with Docker, you need to extend the official n8n image with a custom image that includes your module.
- Set the `NODE_FUNCTION_ALLOW_BUILTIN` and `NODE_FUNCTION_ALLOW_EXTERNAL` environment variables to allow importing modules.

## Using global variables

Sometimes you may wish to set and retrieve simple global data related to a workflow across and within executions. For example, you may wish to include the date of the previous report when compiling a report with a list of project updates.

To set, update, and retrieve data directly to a workflow, use the static data functions within your code. You can manage data either globally or tied to specific nodes.

---

# Compare Datasets

The Compare Datasets node helps you compare data from two input streams.

## Node parameters

1. Decide which fields to compare. In **Input A Field**, enter the name of the field you want to use from input stream A. In **Input B Field**, enter the name of the field you want to use from input stream B.
2. **Optional:** You can compare by multiple fields. Select **Add Fields to Match** to set up more comparisons.
3. Choose how to handle differences between the datasets. In **When There Are Differences**, select one of the following:
  - **Use Input A Version** to treat input stream A as the source of truth.
  - **Use Input B Version** to treat input stream B as the source of truth.
  - **Use a Mix of Versions** to use different inputs for different fields.
    - Use **Prefer** to select either **Input A Version** or **Input B Version** as the main source of truth.
    - Enter input fields that are exceptions to **For Everything Except** to pull from the other input source. To add multiple input fields, enter a comma-separated list.
  - **Include Both Versions** to include both input streams in the output, which may make the structure more complex.
4. Decide whether to use **Fuzzy Compare**. When turned on, the comparison will tolerate small type differences when comparing fields. For example, the number 3 and the string 3 are treated as the same with **Fuzzy Compare** turned on, but wouldn't be treated the same with it turned off.

## Understand item comparison

Item comparison is a two stage process:

1. n8n checks if the values of the fields you selected to compare match across both inputs.
2. If the fields to compare match, n8n then compares all fields within the items, to determine if the items are the same or different.

## Node options

Use the node **Options** to refine your comparison or tweak comparison behavior.

### Fields to Skip Comparing

Enter field names that you want to ignore in the comparison.

For example, if you compare the two datasets below using `person.language` as the **Fields to Match**, n8n returns them as different. If you add `person.name` to **Fields to Skip Comparing**, n8n

returns them as matching.

```
// Input 1
[
 {
 "person": {
 "name": "Stefan",
 "language": "de"
 }
 },
 {
 "person": {
 "name": "Jim",
 "language": "en"
 }
 },
 {
 "person": {
 "name": "Hans",
 "language": "de"
 }
 }
]
// Input 2
[
 {
 "person": {
 "name": "Sara",
 "language": "de"
 }
 },
 {
 "person": {
 "name": "Jane",
 "language": "en"
 }
 },
 {
 "person": {
 "name": "Harriet",
 "language": "de"
 }
 }
]
```

## Disable Dot Notation

Whether to disallow referencing child fields using `parent.child` in the field name (turned on) or allow it (turned off, default).

## Multiple Matches

Choose how to handle duplicate data. The default is **Include All Matches**. You can choose **Include First Match Only**.

For example, given these two datasets:

```
// Input 1
[
 {
 "fruit": {
 "type": "apple",
 "color": "red"
 }
 },
 {
 "fruit": {
 "type": "apple",
 "color": "red"
 }
 },
 {
 "fruit": {
 "type": "banana",
 "color": "yellow"
 }
 }
]
// Input 2
[
 {
 "fruit": {
 "type": "apple",
 "color": "red"
 }
 },
 {
 "fruit": {
 "type": "apple",
 "color": "red"
 }
 },
 {
 "fruit": {
 "type": "banana",
 "color": "yellow"
 }
 }
]
```

n8n returns three items in the **Same Branch** tab. The data is the same in both branches.

If you select **Include First Match Only**, n8n returns two items, in the **Same Branch** tab. The data is the same in both branches, but n8n only returns the first occurrence of the matching “apple” items.

## Understand the output

There are four output options:

- **In A only Branch:** Contains data that occurs only in the first input.
- **Same Branch:** Contains data that's the same in both inputs.
- **Different Branch:** Contains data that's different between inputs.
- **In B only Branch:** Contains data that occurs only in the second output.

## Templates and examples

---

# Compression

Use the Compression node to compress and decompress files.  
Supports Zip and Gzip formats.

## Node parameters

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

The node parameters depend on which **Operation** you select. Choose to:

- **Compress:** Create a compressed file from your input data.
- **Decompress:** Decompress an existing compressed file.

Refer to the sections below for parameters specific to each **Operation**.

### Compress

- **Input Binary Field(s):** Enter the name of the fields in the input data that contain the binary files you want to compress. To compress more than one file, use a comma-separated list.
- **Output Format:** Choose whether to format the compressed output as **Zip** or **Gzip**.
- **File Name:** Enter the name of the zip file the node creates.
- **Put Output File in Field:** Enter the name of the field in the output data to contain the file.

### Decompress

- **Put Output File in Field:** Enter the name of the fields in the input data that contain the binary files you want to decompress. To decompress more than one file, use a comma-separated list.
- **Output Prefix:** Enter a prefix to add to the output file name.

## Templates and examples

---

# Chat Trigger node

Use the Chat Trigger node when building AI workflows for chatbots and other chat interfaces. You can configure how users access the chat, using one of n8n's provided interfaces, or your own. You can add authentication.

You must connect either an agent or chain root node.

## Node parameters

### Make Chat Publicly Available

Set whether the chat should be publicly available (turned on) or only available through the manual chat interface (turned off).

Leave this turned off while you're building the workflow. Turn it on when you're ready to publish the workflow and allow users to access the chat.

### Mode

Choose how users access the chat. Select from:

- **Hosted Chat:** Use n8n's hosted chat interface. Recommended for most users because you can configure the interface using the node options and don't have to do any other setup.
- **Embedded Chat:** This option requires you to create your own chat interface. You can use n8n's chat widget or build your own. Your chat interface must call the webhook URL shown in **Chat URL** in the node.

### Authentication

Choose whether and how to restrict access to the chat. Select from:

- **None:** The chat doesn't use authentication. Anyone can use the chat.
- **Basic Auth:** The chat uses basic authentication.
  - Select or create a **Credential for Basic Auth** with a username and password. All users must use the same username and password.
- **n8n User Auth:** Only users logged in to an n8n account can use the chat.

### Initial Message(s)

This parameter's only available if you're using **Hosted Chat**. Use it to configure the message the n8n chat interface displays when the user arrives on the page.

## Node options

Available options depend on the chat mode.

### Hosted chat options

### **Allowed Origin (CORS)**

Set the origins that can access the chat URL. Enter a comma-separated list of URLs allowed for cross-origin non-preflight requests.

Use \* (default) to allow all origins.

### **Input Placeholder, Title, and Subtitle**

Enter the text for these elements in the chat interface.

??? Details “View screenshot” [Image: Customizable text elements]

### **Load Previous Session**

Select whether to load chat messages from a previous chat session.

If you select any option other than **Off**, you must connect the Chat trigger and the Agent you’re using to a memory sub-node. The memory connector on the Chat trigger appears when you set **Load Previous Session** to **From Memory**. n8n recommends connecting both the Chat trigger and Agent to the same memory sub-node, as this ensures a single source of truth for both nodes.

??? Details “View screenshot” [Image: Connect nodes to memory]

### **Response Mode**

Use this option when building a workflow with steps after the agent or chain that’s handling the chat. Choose from:

- **When Last Node Finishes:** The Chat Trigger node returns the response code and the data output from the last node executed in the workflow.
- **Using Response Nodes:** The Chat Trigger node responds as defined in a Respond to Chat node or Respond to Webhook node. In this response mode, the Chat Trigger will solely show messages as defined in these nodes and not output the data from the last node executed in the workflow.
- **Streaming response:** Enables real-time data streaming back to the user as the workflow processes. Requires nodes with streaming support in the workflow (for example, the AI agent node).

### **Require Button Click to Start Chat**

Set whether to display a **New Conversation** button on the chat interface (turned on) or not (turned off).

??? Details “View screenshot” [Image: New Conversation button]

### **Embedded chat options**

#### **Allowed Origin (CORS)**

Set the origins that can access the chat URL. Enter a comma-separated list of URLs allowed for cross-origin non-preflight requests.

Use \* (default) to allow all origins.

### Load Previous Session

Select whether to load chat messages from a previous chat session.

If you select any option other than **Off**, you must connect the Chat trigger and the Agent you're using to a memory sub-node. The memory connector on the Chat trigger appears when you set **Load Previous Session** to **From Memory**. n8n recommends connecting both the Chat trigger and Agent to the same memory sub-node, as this ensures a single source of truth for both nodes.

??? Details “View screenshot” [Image: Connect nodes to memory]

### Response Mode

Use this option when building a workflow with steps after the agent or chain that's handling the chat. Choose from:

- **When Last Node Finishes:** The Chat Trigger node returns the response code and the data output from the last node executed in the workflow.
- **Using Response Nodes:** The Chat Trigger node responds as defined in a Respond to Chat node or Respond to Webhook node. In this response mode, the Chat Trigger will solely show messages as defined in these nodes and not output the data from the last node executed in the workflow.
- **Streaming response:** Enables real-time data streaming back to the user as the workflow processes. Requires nodes with streaming support enabled.

## Templates and examples

### Related resources

-8<- “ snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

## Set the chat response manually

You need to manually set the chat response when you don't want to directly send the output of an Agent or Chain node to the user. Instead, you want to take the output of an Agent or Chain node and modify it or do something else with it before sending it back to the user.

In a basic workflow, the Agent and Chain nodes output a parameter named either `output` or `text`, and the Chat trigger sends the value of this parameter to the user as the chat response.

If you need to manually create the response sent to the user, you must create a parameter named either `text` or `output`. If you use a different parameter name, the Chat trigger sends the entire object as its

response, not just the value.

## Common issues

For common questions or issues and suggested solutions, refer to [Common Issues](#).

---

## Chat Trigger node common issues

Here are some common errors and issues with the [Chat Trigger node](#) and steps to resolve or troubleshoot them.

### Pass data from a website to an embedded Chat Trigger node

When [embedding](#) the Chat Trigger node in a website, you might want to pass extra information to the Chat Trigger. For example, passing a user ID stored in a site cookie.

To do this, use the `metadata` field in the JSON object you pass to the `createChat` function in your embedded chat window:

```
createChat({
 webhookUrl: 'YOUR_PRODUCTION_WEBHOOK_URL',
 metadata: {
 'YOUR_KEY': 'YOUR_DATA'
 }
});
```

The `metadata` field can contain arbitrary data that will appear in the Chat Trigger output alongside other output data. From there, you can query and process the data from downstream nodes as usual using n8n's [data processing features](#).

### Chat Trigger node doesn't fetch previous messages

When you configure a Chat Trigger node, you might experience problems fetching previous messages if you aren't careful about how you configure session loading. This often manifests as a `workflow could not be started!` error.

In Chat Triggers, the **Load Previous Session** option retrieves previous chat messages for a session using the `sessionId`. When you set the **Load Previous Session** option to **From memory**, it's almost always best to [connect the same memory node](#) to both the Chat Trigger and the Agent in your workflow:

1. In your **Chat Trigger** node, set the **Load Previous Session** option to **From Memory**. This is only visible if you've made the chat publicly available.
2. Attach a **Simple Memory** node to the **Memory** connector.
3. Attach the same **Simple Memory** node to **Memory** connector of

- your **Agent**.
4. In the **Simple Memory** node, set **Session ID** to **Connected Chat Trigger Node**.

One instance where you may want to attach separate memory nodes to your Chat Trigger and the Agent is if you want to set the **Session ID** in your memory node to **Define below**.

If you're retrieving the session ID from an expression, the same expression must work for each of the nodes attached to it. If the expression isn't compatible with each of the nodes that need memory, you might need to use separate memory nodes so you can customize the expression for the session ID on a per-node basis.

---

## Convert to File

Use the Convert to File node to take input data and output it as a file. This converts the input JSON data into a binary format.

### Operations

- [Convert to CSV](#)
- [Convert to HTML](#)
- [Convert to ICS](#)
- [Convert to JSON](#)
- [Convert to ODS](#)
- [Convert to RTF](#)
- [Convert to Text File](#)
- [Convert to XLS](#)
- [Convert to XLSX](#)
- [Move Base64 String to File](#)

Node parameters and options depend on the operation you select.

### Convert to CSV

Configure the node for this operation with the **Put Output File in Field** parameter. Enter the name of the field in the output data to contain the file.

#### Convert to CSV options

You can also configure this operation with these **Options**:

- **File Name:** Enter the file name for the generated output file.
- If the first row of the file contains header names, turn on the **Header Row** option.

### Convert to HTML

Configure the node for this operation with the **Put Output File in Field** parameter. Enter the name of the field in the output data to contain the file.

## Convert to HTML options

You can also configure this operation with these **Options**:

- **File Name:** Enter the file name for the generated output file.
- If the first row of the file contains header names, turn on the **Header Row** option.

## Convert to ICS

- **Put Output File in Field.** Enter the name of the field in the output data to contain the file.
- **Event Title:** Enter the title for the event.
- **Start:** Enter the date and time the event will start. All-day events ignore the time.
- **End:** Enter the date and time the event will end. All-day events ignore the time. If unset, the node uses the start date.
- **All Day:** Select whether the event is an all day event (turned on) or not (turned off).

## Convert to ICS options

You can also configure this operation with these **Options**:

- **File Name:** Enter the file name for the generated output file.
- **Attendees:** Use this option to add attendees to the event. For each attendee, add:
  - **Name**
  - **Email**
  - **RSVP:** Select whether the attendee needs to confirm attendance (turned on) or doesn't (turned off).
- **Busy Status:** Use this option to set the busy status for Microsoft applications like Outlook. Choose from:
  - **Busy**
  - **Tentative**
- **Calendar Name:** For Apple and Microsoft calendars, enter the calendar name for the event.
- **Description:** Enter an event description.
- **Geolocation:** Enter the **Latitude** and **Longitude** for the event's location.
- **Location:** Enter the event's intended venue/location.
- **Recurrence Rule:** Enter a rule to define the repeat pattern of the event (RRULE). Generate rules using the [iCalendar.org RRULE Tool](#).
- **Organizer:** Enter the organizer's **Name** and **Email**.
- **Sequence:** If you're sending an update for an event with the same universally unique ID (UID), enter the revision sequence number.
- **Status:** Set the status of the event. Choose from:
  - **Confirmed**
  - **Cancelled**
  - **Tentative**
- **UID:** Enter a universally unique ID (UID) for the event. The UID should be globally unique. The node automatically generates a UID if you don't enter one.
- **URL:** Enter a URL associated with the event.
- **Use Workflow Timezone:** Whether to use UTC time zone (turned off) or the workflow's timezone (turned on). Set the workflow's timezone in the [Workflow Settings](#).

## Convert to JSON

Choose the best output **Mode** for your needs from these options:

- **All Items to One File**: Send all input items to a single file.
- **Each Item to Separate File**: Create a file for every input item.

### Convert to JSON options

You can also configure this operation with these **Options**:

- **File Name**: Enter the file name for the generated output file.
- **Format**: Choose whether to format the JSON for easier reading (turned on) or not (turned off).
- **Encoding**: Choose the character set to use to encode the data. The default is **utf8**.

## Convert to ODS

Configure the node for this operation with the **Put Output File in Field** parameter. Enter the name of the field in the output data to contain the file.

### Convert to ODS options

You can also configure this operation with these **Options**:

- **File Name**: Enter the file name for the generated output file.
- **Compression**: Choose whether to compress and reduce the file's output size.
- **Header Row**: Turn on if the first row of the file contains header names.
- **Sheet Name**: Enter the Sheet Name to create in the spreadsheet.

## Convert to RTF

Configure the node for this operation with the **Put Output File in Field** parameter. Enter the name of the field in the output data to contain the file.

### Convert to RFT options

You can also configure this operation with these **Options**:

- **File Name**: Enter the file name for the generated output file.
- If the first row of the file contains header names, turn on the **Header Row** option.

## Convert to Text File

Enter the name of the **Text Input Field** that contains a string to convert to a file. Use dot-notation for deep fields, for example `level1.level2.currentKey`.

### Convert to Text File options

You can also configure this operation with these **Options**:

- **File Name:** Enter the file name for the generated output file.
- **Encoding:** Choose the character set to use to encode the data. The default is **utf8**.

## Convert to XLS

Configure the node for this operation with the **Put Output File in Field** parameter. Enter the name of the field in the output data to contain the file.

### Convert to XLS options

You can also configure this operation with these **Options**:

- **File Name:** Enter the file name for the generated output file.
- **Header Row:** Turn on if the first row of the file contains header names.
- **Sheet Name:** Enter the Sheet Name to create in the spreadsheet.

## Convert to XLSX

Configure the node for this operation with the **Put Output File in Field** parameter. Enter the name of the field in the output data to contain the file.

### Convert to XLSX options

You can also configure this operation with these **Options**:

- **File Name:** Enter the file name for the generated output file.
- **Compression:** Choose whether to compress and reduce the file's output size.
- **Header Row:** Turn on if the first row of the file contains header names.
- **Sheet Name:** Enter the Sheet Name to create in the spreadsheet.

## Move Base64 String to File

Enter the name of the **Base64 Input Field** that contains the Base64 string to convert to a file. Use dot-notation for deep fields, for example `level1.level2.currentKey`.

### Move Base64 String to File options

You can also configure this operation with these **Options**:

- **File Name:** Enter the file name for the generated output file.
- **MIME Type:** Enter the MIME type of the output file. Refer to [Common MIME types](#) for a list of common MIME types and the file extensions they relate to.

## Templates and examples

---

# Crypto

Use the Crypto node to encrypt data in workflows.

## Actions

- Generate a random string
- Hash a text or file in a specified format
- Hmac a text or file in a specified format
- Sign a string using a private key

## Node parameters

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

Node parameters depend on the action you select.

### Generate parameters

- **Property Name:** Enter the name of the property to write the random string to.
- **Type:** Select the encoding type to use to generate the string. Choose from:
  - ASCII
  - BASE64
  - HEX
  - UUID

### Hash parameters

- **Type:** Select the hash type to use. Choose from:
  - MD5
  - SHA256
  - SHA3-256
  - SHA3-384
  - SHA3-512
  - SHA385
  - SHA512
- **Binary File:** Turn this parameter on if the data you want to hash is from a binary file.
  - **Value:** If you turn off **Binary File**, enter the value you want to hash.
  - **Binary Property Name:** If you turn on **Binary File**, enter the name of the binary property that contains the data you want to hash.
- **Property Name:** Enter the name of the property you want to write the hash to.
- **Encoding:** Select the encoding type to use. Choose from:
  - BASE64
  - HEX

### Hmac parameters

- **Binary File:** Turn this parameter on if the data you want to

encrypt is from a binary file.

- **Value:** If you turn off **Binary File**, enter the value you want to encrypt.
- **Binary Property Name:** If you turn on **Binary File**, enter the name of the binary property that contains the data you want to encrypt.
- **Type:** Select the encryption type to use. Choose from:
  - **MD5**
  - **SHA256**
  - **SHA3-256**
  - **SHA3-384**
  - **SHA3-512**
  - **SHA385**
  - **SHA512**
- **Property Name:** Enter the name of the property you want to write the hash to.
- **Secret:** Enter the secret or secret key used for decoding.
- **Encoding:** Select the encoding type to use. Choose from:
  - **BASE64**
  - **HEX**

## Sign parameters

- **Value:** Enter the value you want to sign.
- **Property Name:** Enter the name of the property you want to write the signed value to.
- **Algorithm Name or ID:** Choose an algorithm name from the list or specify an ID using an expression.
- **Encoding:** Select the encoding type to use. Choose from:
  - **BASE64**
  - **HEX**
- **Private Key:** Enter a private key to use when signing the string.

## Templates and examples

---

## Data table

Use the Data Table node to permanently save data across workflow executions in a table format. It provides functionality to perform various data operations on stored data. See [Data tables](#).

## Node parameters

### Resource

Select the resource on which you want to operate.

- Rows

### Operations

Select the operation you want to run on the resource:

- **Delete:** Delete one or more rows.
  - **Dry Run:** Simulate a deletion before finalizing it. If you switch on this option, n8n returns the rows that will be deleted by the operation. Default state is off.
- **Get:** Get one or more rows from your table based on defined filters.
  - **Limit:** The number of rows you want to return, specified as a number. Default value is 50.
  - **Return all:** Switch on to return all data. Default value is off.
- **If Row Exists:** Specify a set of conditions to match input items that exist in the data table.
- **If Row Does Not Exist:** Specify a set of conditions to match input items that don't exist in the data table.
- **Insert:** Insert rows into an existing table.
  - **Optimize Bulk:** Optimize the speed of insertions when working with many rows. If you switch on this option, n8n won't return the data that was inserted. Default state is off.
- **Update:** Update one or more rows.
- **Upsert:** Upsert one or more rows. If the row exists, it's updated; otherwise, a new row is created.

## Related resources

[Data tables](#) explains how to create and manage data tables.

---

## Date & Time

The Date & Time node manipulates date and time data and convert it to different formats.

-8<- “\_snippets/integrations/builtin/core-nodes/schedule/timezone-settings.md”

## Operations

- **Add to a Date:** Add a specified amount of time to a date.
- **Extract Part of a Date:** Extract part of a date, such as the year, month, or day.
- **Format a Date:** Transform a date's format to a new format using preset options or a custom expression.
- **Get Current Date:** Get the current date and choose whether to include the current time or not. Useful for triggering other flows and conditional logic.
- **Get Time Between Dates:** Calculate the amount of time in specific units between two dates.
- **Round a Date:** Round a date up or down to the nearest unit of your choice, such as month, day, or hour.
- **Subtract From a Date:** Subtract a specified amount of time from a date.

Refer to the sections below for parameters and options specific to each operation.

## Add to a Date

Configure the node for this operation using these parameters:

- **Date to Add To:** Enter the date you want to change.
- **Time Unit to Add:** Select the time unit for the **Duration** parameter.
- **Duration:** Enter the number of time units to add to the date.
- **Output Field Name:** Enter the name of the field to output the new date to.

### Add to a Date options

This operation has one option: **Include Input Fields**. If you'd like to include all of the input fields in the output, turn this option on. If turned off, only the **Output Field Name** and its contents are output.

## Extract Part of a Date

Configure the node for this operation using these parameters:

- **Date:** Enter the date you want to round or extract part of.
- **Part:** Select the part of the date you want to extract. Choose from:
  - **Year**
  - **Month**
  - **Week**
  - **Day**
  - **Hour**
  - **Minute**
  - **Second**
- **Output Field Name:** Enter the name of the field to output the extracted date part to.

### Extract Part of a Date options

This operation has one option: **Include Input Fields**. If you'd like to include all of the input fields in the output, turn this option on. If turned off, only the **Output Field Name** and its contents are output.

## Format a Date

Configure the node for this operation using these parameters:

- **Date:** Enter the date you want to format.
- **Format:** Select the format you want to change the date to. Choose from:
  - **Custom Format:** Enter your own custom format using Luxon's special tokens. Tokens are case-sensitive.
  - **MM/DD/YYYY:** For 4 September 1986, this formats the date as 09/04/1986.
  - **YYYY/MM/DD:** For 4 September 1986, this formats the date as 1986/09/04.
  - **MMMM DD YYYY:** For 4 September 1986, this formats the date as September 04 1986.
  - **MM-DD-YYYY:** For 4 September 1986, this formats the date as

- 09-04-1986.
- **YYYY-MM-DD:** For 4 September 1986, this formats the date as 1986-09-04.
- **Output Field Name:** Enter the name of the field to output the formatted date to.

## Format a Date options

This operation includes these options:

- **Include Input Fields:** If you'd like to include all of the input fields in the output, turn this option on. If turned off, only the **Output Field Name** and its contents are output.
- **From Date Format:** If the node isn't recognizing the **Date** format correctly, enter the format for that **Date** here so the node can process it properly. Use Luxon's special tokens to enter the format. Tokens are case-sensitive
- **Use Workflow Timezone:** Whether to use the input's time zone (turned off) or the workflow's timezone (turned on).

## Get Current Date

Configure the node for this operation using these parameters:

- **Include Current Time:** Choose whether to include the current time (turned on) or to set the time to midnight (turned off).
- **Output Field Name:** Enter the name of the field to output the current date to.

## Get Current Date options

This operation includes these options:

- **Include Input Fields:** If you'd like to include all of the input fields in the output, turn this option on. If turned off, only the **Output Field Name** and its contents are output.
- **Timezone:** Set the timezone to use. If left blank, the node uses the n8n instance's timezone.

## Get Time Between Dates

Configure the node for this operation using these parameters:

- **Start Date:** Enter the earlier date you want to compare.
- **End Date:** Enter the later date you want to compare.
- **Units:** Select the units you want to calculate the time between. You can include multiple units. Choose from:
  - **Year**
  - **Month**
  - **Week**
  - **Day**
  - **Hour**
  - **Minute**
  - **Second**
  - **Millisecond**
- **Output Field Name:** Enter the name of the field to output the

calculated time between to.

## Get Time Between Dates options

The Get Time Between Dates operation includes the **Include Input Fields** option as well as an **Output as ISO String** option. If you leave this option off, each unit you selected will return its own time difference calculation, for example:

```
timeDifference
years : 1
months : 3
days : 13
```

If you turn on the **Output as ISO String** option, the node formats the output as a single ISO duration string, for example: P1Y3M13D.

ISO duration format displays a format as P<n>Y<n>M<n>DT<n>H<n>M<n>S. <n> is the number for the unit after it.

- P = period (duration). It begins all ISO duration strings.
- Y = years
- M = months
- W = weeks
- D = days
- T = delineator between dates and times, used to avoid confusion between months and minutes
- H = hours
- M = minutes
- S = seconds

Milliseconds don't get their own unit, but instead are decimal seconds. For example, 2.1 milliseconds is 0.0021S.

## Round a Date

Configure the node for this operation using these parameters:

- **Date:** Enter the date you'd like to round.
- **Mode:** Choose whether to **Round Down** or **Round Up**.
- **To Nearest:** Select the unit you'd like to round to. Choose from:
  - Year
  - Month
  - Week
  - Day
  - Hour
  - Minute
  - Second
- **Output Field Name:** Enter the name of the field to output the rounded date to.

## Round a Date options

This operation has one option: **Include Input Fields**. If you'd like to include all of the input fields in the output, turn this option on. If turned off, only the **Output Field Name** and its contents are output.

## Subtract From a Date

Configure the node for this operation using these parameters:

- **Date to Subtract From:** Enter the date you'd like to subtract from.
- **Time Unit to Subtract:** Select the unit for the **Duration** amount you want to subtract.
- **Duration:** Enter the amount of the time units you want to subtract from the **Date to Subtract From**.
- **Output Field Name:** Enter the name of the field to output the rounded date to.

### Subtract From a Date options

This operation has one option: **Include Input Fields**. If you'd like to include all of the input fields in the output, turn this option on. If turned off, only the **Output Field Name** and its contents are output.

## Templates and examples

### Related resources

The Date & Time node uses [Luxon](#). You can also use Luxon in the [Code](#) node and [expressions](#). Refer to [Date and time with Luxon](#) for more information.

### Supported date formats

n8n supports all date formats [supported by Luxon](#). Tokens are case-sensitive.

---

## Debug Helper

Use the Debug Helper node to trigger different error types or generate random datasets to help test n8n workflows.

## Operations

Define the operation by selecting the **Category**:

- **Do Nothing:** Don't do anything.
- **Throw Error:** Throw an error with the specified type and message.
- **Out Of Memory:** Generate a specific memory size to simulate being out of memory.
- **Generate Random Data:** Generate some random data in a selected format.

## Node parameters

The node parameters depend on the **Category** selected. The **Do Nothing** Category has no other parameters.

## Throw Error

- **Error Type:** Select the type of error to throw. Choose from:
  - **NodeApiError**
  - **NodeOperationError**
  - **Error**
- **Error Message:** Enter the error message to throw.

## Out Of Memory

The Out of Memory Category adds one parameter, the **Memory Size to Generate**. Enter the approximate amount of memory to generate.

## Generate Random Data

- **Data Type:** Choose the type of random data you'd like to generate. Options include:
  - **Address**
  - **Coordinates**
  - **Credit Card**
  - **Email**
  - **IPv4**
  - **IPv6**
  - **MAC**
  - **Nanoids:** If you select this data type, you'll also need to enter:
    - **Nanoid Alphabet:** The alphabet the generator will use to generate the nanoids.
    - **Nanoid Length:** The length of each nanoid.
  - **URL**
  - **User Data**
  - **UUID**
  - **Version**
- **Seed:** If you'd like to generate the data using a specific seed, enter it here. This ensures the data gets generated consistently. If you'd rather use random data generation, leave this field empty.
- **Number of Items to Generate:** Enter the number of random items you'd like to generate.
- **Output as Single Array:** Whether to generate the data as a single array (turned on) or multiple items (turned off).

## Templates and examples

---

## Edit Fields (Set)

Use the Edit Fields node to set workflow data. This node can set new data as well as overwrite data that already exists. This node is crucial in workflows which expect incoming data from previous nodes, such as when inserting values to Google Sheets or databases.

## Node parameters

These are the settings and options available in the Edit Fields node.

### Mode

You can either use **Manual Mapping** to edit fields using the GUI or **JSON Output** to write JSON that n8n adds to the input data.

### Fields to Set

If you select **Mode > Manual Mapping**, you can configure the fields by dragging and dropping values from **INPUT**.

The default behavior when you drag a value is:

- n8n sets the value's name as the field name.
- The field value contains an expression which accesses the value.

If you don't want to use expressions:

1. Hover over a field. n8n displays the **Fixed | Expressions** toggle.
2. Select **Fixed**.

You can do this for both the name and value of the field.

[Image: A gif showing the drag and drop action, as well as changing a field to fixed]

### Keep Only Set Fields

Enable this to discard any input data that you don't use in **Fields to Set**.

### Include in Output

Choose which input data to include in the node's output data.

## Node options

Use these options to customize the behavior of the node.

### Include Binary Data

If the input data includes binary data, choose whether to include it in the Edit Fields node's output data.

### Ignore Type Conversion Errors

Manual Mapping only.

Enabling this allows n8n to ignore some data type errors when mapping fields.

### Support Dot Notation

By default, n8n supports dot notation.

For example, when using manual mapping, the node follows the dot notation for the **Name** field. That means if you set the name in the **Name** field as `number.one` and the value in the **Value** field as `20`, the resulting JSON is:

```
{ "number": { "one": 20} }
```

You can prevent this behavior by selecting **Add Option > Support Dot Notation**, and setting the **Dot Notation** field to off. Now the resulting JSON is:

```
{ "number.one": 20 }
```

## Templates and examples

### Arrays and expressions in JSON Output mode

You can use arrays and expressions when creating your JSON Output.

For example, given this input data generated by the Customer Datastore node:

```
[
 {
 "id": "23423532",
 "name": "Jay Gatsby",
 "email": "gatsby@west-egg.com",
 "notes": "Keeps asking about a green light???",
 "country": "US",
 "created": "1925-04-10"
 },
 {
 "id": "23423533",
 "name": "José Arcadio Buendía",
 "email": "jab@macondo.co",
 "notes": "Lots of people named after him. Very confusing",
 "country": "CO",
 "created": "1967-05-05"
 },
 {
 "id": "23423534",
 "name": "Max Sendak",
 "email": "info@in-and-out-of-weeks.org",
 "notes": "Keeps rolling his terrible eyes",
 "country": "US",
 "created": "1963-04-09"
 },
 {
 "id": "23423535",
 "name": "Zaphod Beeblebrox",
 "email": "captain@heartofgold.com",
 "notes": "Felt like I was talking to more than one person",
 "country": null,
 "created": "1979-10-12"
 },
```

```
{
 "id": "23423536",
 "name": "Edmund Pevensie",
 "email": "edmund@narnia.gov",
 "notes": "Passionate sailor",
 "country": "UK",
 "created": "1950-10-16"
}
]
```

Add the following JSON in the **JSON Output** field, with **Include in Output** set to **All Input Fields**:

```
{
 "newKey": "new value",
 "array": [{"$json.id"}, "{$json.name}"],
 "object": {
 "innerKey1": "new value",
 "innerKey2": "{$json.id}",
 "innerKey3": "{$json.name}",
 }
}
```

You get this output:

```
[
 {
 "id": "23423532",
 "name": "Jay Gatsby",
 "email": "gatsby@west-egg.com",
 "notes": "Keeps asking about a green light??",
 "country": "US",
 "created": "1925-04-10",
 "newKey": "new value",
 "array": [
 23423532,
 "Jay Gatsby"
],
 "object": {
 "innerKey1": "new value",
 "innerKey2": "23423532",
 "innerKey3": "Jay Gatsby"
 }
 },
 {
 "id": "23423533",
 "name": "José Arcadio Buendía",
 "email": "jab@macondo.co",
 "notes": "Lots of people named after him. Very confusing",
 "country": "CO",
 "created": "1967-05-05",
 "newKey": "new value",
 "array": [
 23423533,
 "José Arcadio Buendía"
],
 "object": {
 "innerKey1": "new value",
 "innerKey2": "23423533",
 "innerKey3": "José Arcadio Buendía"
 }
 }
]
```

```
},
{
 "id": "23423534",
 "name": "Max Sendak",
 "email": "info@in-and-out-of-weeks.org",
 "notes": "Keeps rolling his terrible eyes",
 "country": "US",
 "created": "1963-04-09",
 "newKey": "new value",
 "array": [
 23423534,
 "Max Sendak"
],
 "object": {
 "innerKey1": "new value",
 "innerKey2": "23423534",
 "innerKey3": "Max Sendak"
 }
},
{
 "id": "23423535",
 "name": "Zaphod Beeblebrox",
 "email": "captain@heartofgold.com",
 "notes": "Felt like I was talking to more than one person",
 "country": null,
 "created": "1979-10-12",
 "newKey": "new value",
 "array": [
 23423535,
 "Zaphod Beeblebrox"
],
 "object": {
 "innerKey1": "new value",
 "innerKey2": "23423535",
 "innerKey3": "Zaphod Beeblebrox"
 }
},
{
 "id": "23423536",
 "name": "Edmund Pevensie",
 "email": "edmund@narnia.gov",
 "notes": "Passionate sailor",
 "country": "UK",
 "created": "1950-10-16",
 "newKey": "new value",
 "array": [
 23423536,
 "Edmund Pevensie"
],
 "object": {
 "innerKey1": "new value",
 "innerKey2": "23423536",
 "innerKey3": "Edmund Pevensie"
 }
}
]
```

---

# Edit Image

Use the Edit Image node to manipulate and edit images.

## Operations

- Add a **Blur** to the image to reduce sharpness
- Add a **Border** to the image
- **Composite** an image on top of another image
- **Create** a new image
- **Crop** the image
- **Draw** on an image
- **Get Information** about the image
- **Multi Step** perform multiple operations on the image
- **Resize**: Change the size of the image
- **Rotate** the image
- **Shear** image along the X or Y axis
- Add **Text** to the image
- Make a color in image **Transparent**

## Node parameters

The parameters for this node depend on the operation you select.

### Blur parameters

- **Property Name:** Enter the name of the binary property that stores the image data.
- **Blur:** Enter a number to set how strong the blur should be, between 0 and 1000. Higher numbers create blurrier images.
- **Sigma:** Enter a number to set the stigma for the blur, between 0 and 1000. Higher numbers create blurrier images.

Refer to [Node options](#) for optional configuration options.

### Border parameters

- **Property Name:** Enter the name of the binary property that stores the image data.
- **Border Width:** Enter the width of the border.
- **Border Height:** Enter the height of the border.
- **Border Color:** Set the color for the border. You can either enter a hex or select the color swatch to open a color picker.

Refer to [Node options](#) for optional configuration options.

### Composite parameters

- **Property Name:** Enter the name of the binary property that stores the image data. This image is your base image.
- **Composite Image Property:** Enter the name of the binary property that stores image to composite on top of the **Property Name** image.
- **Operator:** Select composite operator, which determines how the

composite works. Options include:

- **Add**
- **Atop**
- **Bumpmap**
- **Copy**
- **Copy Black**
- **Copy Blue**
- **Copy Cyan**
- **Copy Green**
- **Copy Magenta**
- **Copy Opacity**
- **Copy Red**
- **Copy Yellow**
- **Difference**
- **Divide**
- **In**
- **Minus**
- **Multiply**
- **Out**
- **Over**
- **Plus**
- **Subtract**
- **Xor**

- **Position X:** Enter the x axis position (horizontal) of the composite image.
- **Position Y:** Enter the y axis position (vertical) of the composite image.

Refer to [Node options](#) for optional configuration options.

## Create parameters

- **Property Name:** Enter the name of the binary property that stores the image data.
- **Background Color:** Set the background color for the image. You can either enter a hex or select the color swatch to open a color picker.
- **Image Width:** Enter the width of the image.
- **Image Height:** Enter the height of the image.

Refer to [Node options](#) for optional configuration options.

## Crop parameters

- **Property Name:** Enter the name of the binary property that stores the image data.
- **Width:** Enter the width you'd like to crop to.
- **Height:** Enter the height you'd like to crop to.
- **Position X:** Enter the x axis position (horizontal) to start the crop from.
- **Position Y:** Enter the y axis position (vertical) to start the crop from.

Refer to [Node options](#) for optional configuration options.

## Draw parameters

- **Property Name:** Enter the name of the binary property that stores the image data.
- **Primitive:** Select the primitive shape to draw. Choose from:
  - **Circle**
  - **Line**
  - **Rectangle**
- **Color:** Set the color for the primitive. You can either enter a hex or select the color swatch to open a color picker.
- **Start Position X:** Enter the x axis position (horizontal) to start drawing from.
- **Start Position Y:** Enter the y axis position (vertical) to start drawing from.
- **End Position X:** Enter the x axis position (horizontal) to stop drawing at.
- **End Position Y:** Enter the y axis position (vertical) to start drawing at.
- **Corner Radius:** Enter a number to set the corner radius. Adding a corner radius will round the corners of the drawn primitive.

Refer to [Node options](#) for optional configuration options.

## Get Information parameters

For this operation, you only need to add the **Property Name** of the binary property that stores the image data.

Refer to [Node options](#) for optional configuration options.

## Multi Step parameters

- **Property Name:** Enter the name of the binary property that stores the image data.
- **Operations:** Add the operations you want the multi step operation to perform. You can use any of the other operations.

Refer to [Node options](#) for optional configuration options.

## Resize parameters

- **Property Name:** Enter the name of the binary property that stores the image data.
- **Width:** Enter the new width you'd like for the image.
- **Height:** Enter the new height you'd like for the image.
- **Option:** Select how you'd like to resize the image. Choose from:
  - **Ignore Aspect Ratio:** Ignore the aspect ratio and resize to the exact height and width you've entered.
  - **Maximum Area:** The height and width you've entered is the maximum area/size for the image. The image maintains its aspect ratio and won't be larger than the height and/or width you've entered.
  - **Minimum Area:** The height and width you've entered is the minimum area/size for the image. The image maintains its aspect ratio and won't be smaller than the height and/or width you've entered.
  - **Only if Larger:** Resize the image only if it's larger than the width and height you entered. The image maintains its aspect ratio.
  - **Only if Smaller:** Resize the image only if it's smaller than the

width and height you entered. The image maintains its aspect ratio.

- **Percent:** Resize the image using the width and height as percentages of the original image.

Refer to [Node options](#) for optional configuration options.

## Rotate parameters

- **Property Name:** Enter the name of the binary property that stores the image data.
- **Rotate:** Enter the number of degrees to rotate the image, from -360 to 360.
- **Background Color:** Set the background color for the image. You can either enter a hex or select the color swatch to open a color picker. This color is used to fill in the empty background whenever the image is rotated by multiples of 90 degrees. If multiplied of 90 degrees are used for the **Rotate** field, the background color isn't used.

Refer to [Node options](#) for optional configuration options.

## Shear parameters

- **Property Name:** Enter the name of the binary property that stores the image data.
- **Degrees X:** Enter the number of degrees to shear from the x axis.
- **Degrees Y:** Enter the number of degrees to shear from the y axis.

Refer to [Node options](#) for optional configuration options.

## Text parameters

- **Property Name:** Enter the name of the binary property that stores the image data.
- **Text:** Enter the text you'd like to write on the image.
- **Font Size:** Select the font size for the text.
- **Font Color:** Set the font color. You can either enter a hex or select the color swatch to open a color picker.
- **Position X:** Enter the x axis position (horizontal) to begin the text at.
- **Position Y:** Enter the y axis position (vertical) to begin the text at.
- **Max Line Length:** Enter the maximum amount of characters in a line before adding a line break.

Refer to [Node options](#) for optional configuration options.

## Transparent parameters

- **Property Name:** Enter the name of the binary property that stores the image data.
- **Color:** Set the color to make transparent. You can either enter a hex or select the color swatch to open a color picker.

Refer to [Node options](#) for optional configuration options.

## Node options

- **File Name:** Enter the filename of the output file.
- **Format:** Enter the image format of the output file. Choose from:
  - **bmp**
  - **gif**
  - **jpeg**
  - **png**
  - **tiff**
  - **WebP**

The **Text** operation also includes the option for **Font Name or ID**. Select the text font from the dropdown or specify an ID using an [expression](#).

## Templates and examples

---

## Email Trigger (IMAP) node

Use the IMAP Email node to receive emails using an IMAP email server. This node is a trigger node.

### Operations

- Receive an email

### Node parameters

Configure the node using the following parameters.

#### Credential to connect with

Select or create an [IMAP credential](#) to connect to the server with.

#### Mailbox Name

Enter the mailbox from which you want to receive emails.

#### Action

Choose whether you want an email marked as read when n8n receives it. **None** will leave it marked unread. **Mark as Read** will mark it as read.

#### Download Attachments

This toggle controls whether to download email attachments (turned on) or not (turned off). Only set this if necessary, since it increases processing.

#### Format

Choose the format to return the message in from these options:

- **RAW**: This format returns the full email message data with body content in the raw field as a base64url encoded string. It doesn't use the payload field.
- **Resolved**: This format returns the full email with all data resolved and attachments saved as binary data.
- **Simple**: This format returns the full email. Don't use it if you want to gather inline attachments.

## Node options

You can further configure the node using these **Options**.

### Custom Email Rules

Enter custom email fetching rules to determine which emails the node fetches.

Refer to [node-imap's search function criteria](#) for more information.

### Force Reconnect Every Minutes

Set an interval in minutes to force reconnection.

## Templates and examples

---

## Error Trigger node

You can use the Error Trigger node to create error workflows. When another linked workflow fails, this node gets details about the failed workflow and the errors, and runs the error workflow.

## Usage

```
-8<- "_snippets/flow-logic/create-set-error-workflow.md"
```

Note the following:

- If a workflow uses the Error Trigger node, you don't have to publish the workflow.
- If a workflow contains the Error Trigger node, by default, the workflow uses itself as the error workflow.
- You can't test error workflows when running workflows manually. The Error Trigger only runs when an automatic workflow errors.

## Templates and examples

## Related resources

You can use the [Stop And Error](#) node to send custom messages to the Error Trigger.

Read more about [Error workflows](#) in n8n workflows.

## Error data

-8<- “\_snippets/integrations/builtin/core-nodes/error-trigger/error-data.md”

---

## Evaluation node

The Evaluation node performs various operations related to [evaluations](#) to validate your AI workflow reliability.

Use the Evaluation node in these scenarios:

- To conditionally execute logic based on whether the workflow is under evaluation
- To write evaluation outcomes back to a Google Sheet dataset or
- To log scoring metrics for your evaluation performance to n8n’s evaluations tab

## Operations

The Evaluation node offers the following operations:

- **Set Outputs:** Write the results of an evaluation back to a data table or Google Sheet dataset.
- **Set Metrics:** Record metrics scoring the evaluation performance to n8n’s **Evaluations** tab.
- **Check If Evaluating:** Branches the workflow execution logic depending on whether the current execution is an evaluation.

The parameters and options available depend on the operation you select.

### Set Outputs

The **Set Outputs** operation has the following parameters:

- **Source:** Select the location to which you want to output the evaluation results. Default value is **Data table**.

Source settings differ depending on **Source** selection.

- When **Source** is **Data table**:
  - **Data table:** Select a data table by name or ID
- When **Source** is **Google Sheets**:
  - **Credential to connect with:** Create or select an existing [Google Sheets credentials](#).
  - **Document Containing Dataset:** Choose the spreadsheet document you want to write the evaluation results to. Usually this is the same document you select in the [Evaluation Trigger](#) node.

- Select **From list** to choose the spreadsheet title from the dropdown list, **By URL** to enter the url of the spreadsheet, or **By ID** to enter the spreadsheetId.
  - You can find the spreadsheetId in a Google Sheets URL: <https://docs.google.com/spreadsheets/d/spreadsheetId/edit#gid=0>.
- **Sheet Containing Dataset:** Choose the sheet you want to write the evaluation results to. Usually this is the same sheet you select in the Evaluation Trigger node.
  - Select **From list** to choose the sheet title from the dropdown list, **By URL** to enter the url of the sheet, **By ID** to enter the sheetId, or **By Name** to enter the sheet title.
  - You can find the sheetId in a Google Sheets URL: [https://docs.google.com/spreadsheets/d/aBC-123\\_xYz/edit#gid=sheetId](https://docs.google.com/spreadsheets/d/aBC-123_xYz/edit#gid=sheetId).

You define the items to write to the data table or Google Sheet in the **Outputs** section. For each output, you set the following:

- **Name:** The Google Sheet column name to write the evaluation results to.
- **Value:** The value to write to the Google Sheet.

## Set Metrics

The **Set Metrics** operation includes a **Metrics to Return** section where you define the metrics to record and track for your evaluations. You can see the metric results in your workflow's **Evaluations** tab.

For each metric you wish to record, you set the following details:

- **Name:** The name to use for the metric.
- **Value:** The numeric value to record. Once you run your evaluation, you can drag and drop values from previous nodes here. Metric values must be numeric.

## Check If Evaluating

The **Check If Evaluating** operation doesn't have any parameters. This operation provides branching output connectors so that you can conditionally execute logic depending on whether the current execution is an evaluation or not.

## Templates and examples

## Related resources

To learn more about n8n evaluations, check out the [evaluations documentation](#)

n8n provides a trigger node for evaluations. You can find the node docs [here](#).

For common questions or issues and suggested solutions, refer to the [evaluations tips and common issues](#) page.