

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
 - Date/Time
 - 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

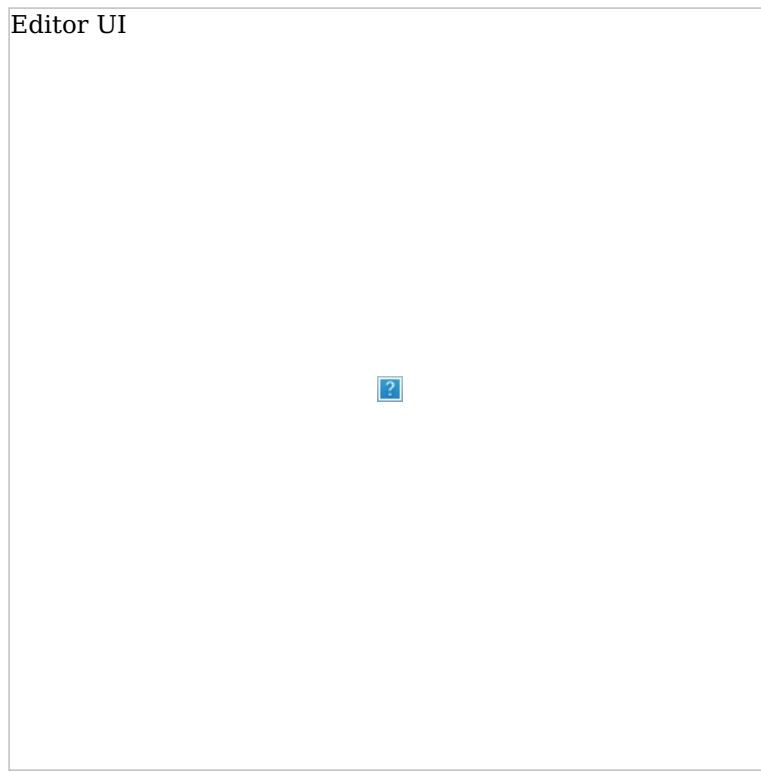
Begin by setting up n8n.

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.

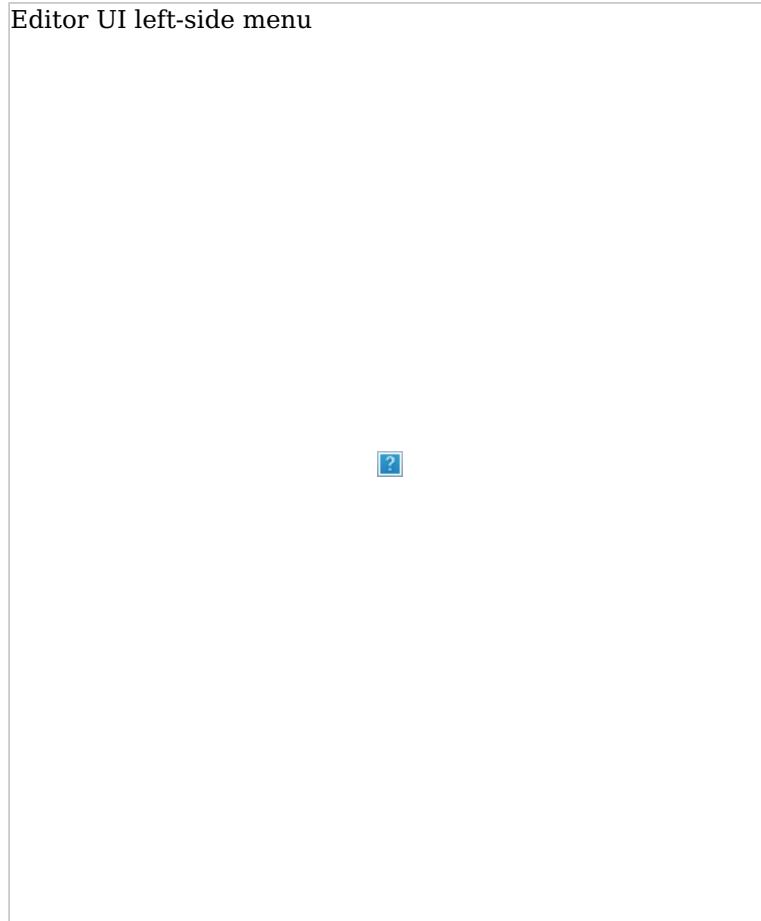
Left-side panel

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

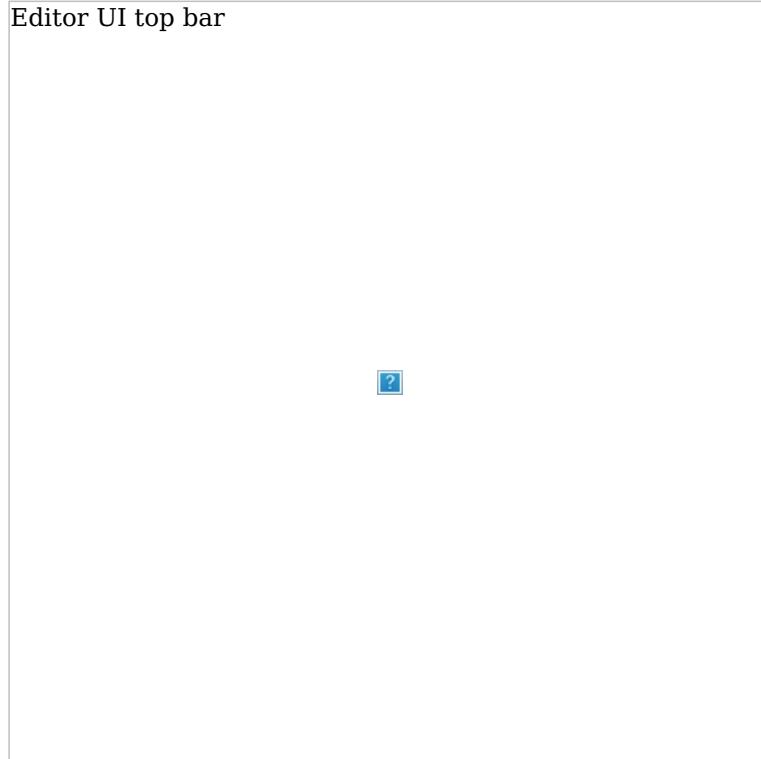
Top bar

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.

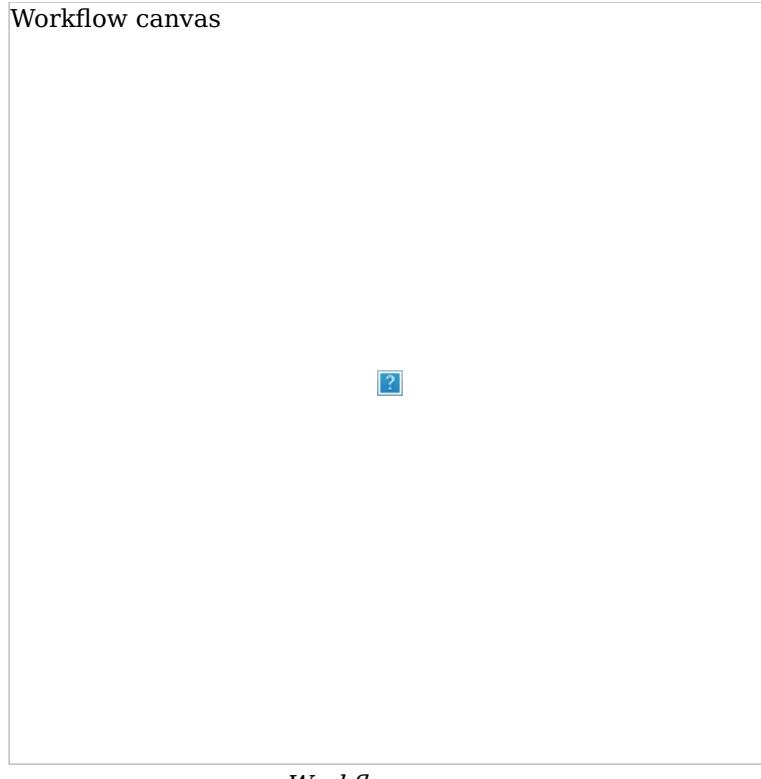


Editor UI top bar

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.

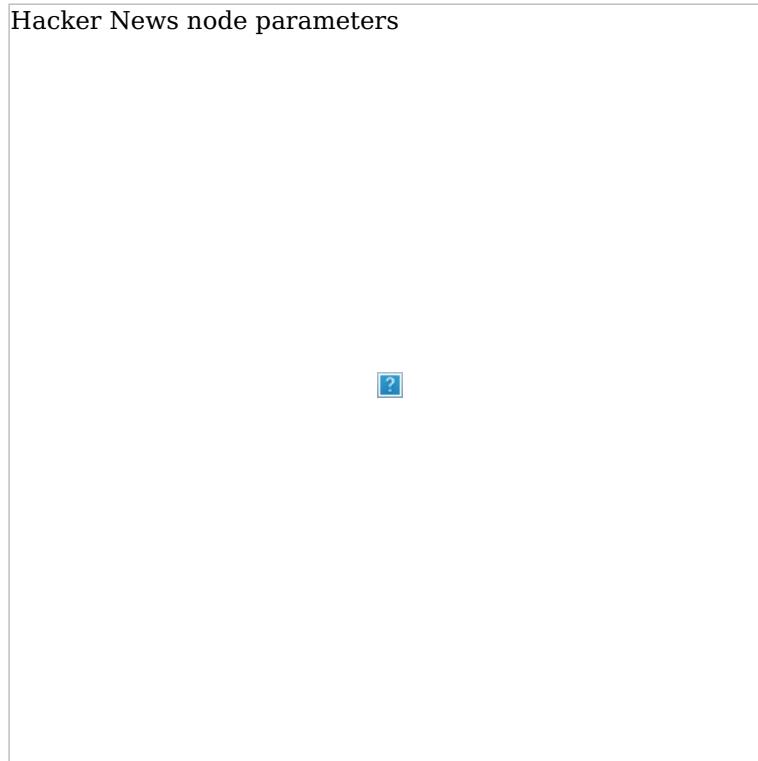
Parameters

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

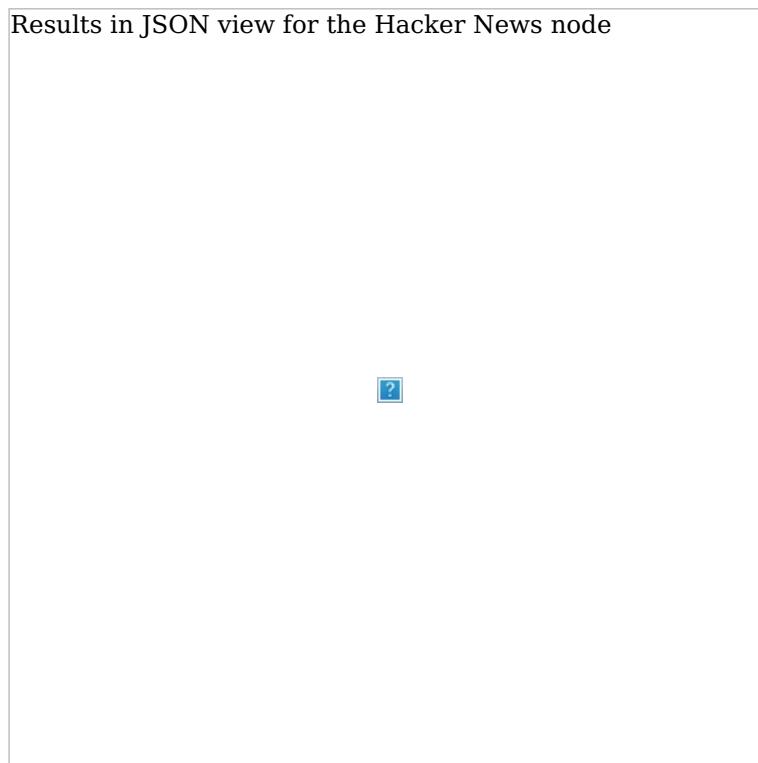
Successfully executed workflow



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.

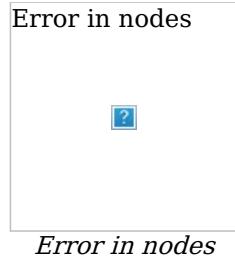
Here's our Hacker News output in JSON view:



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."

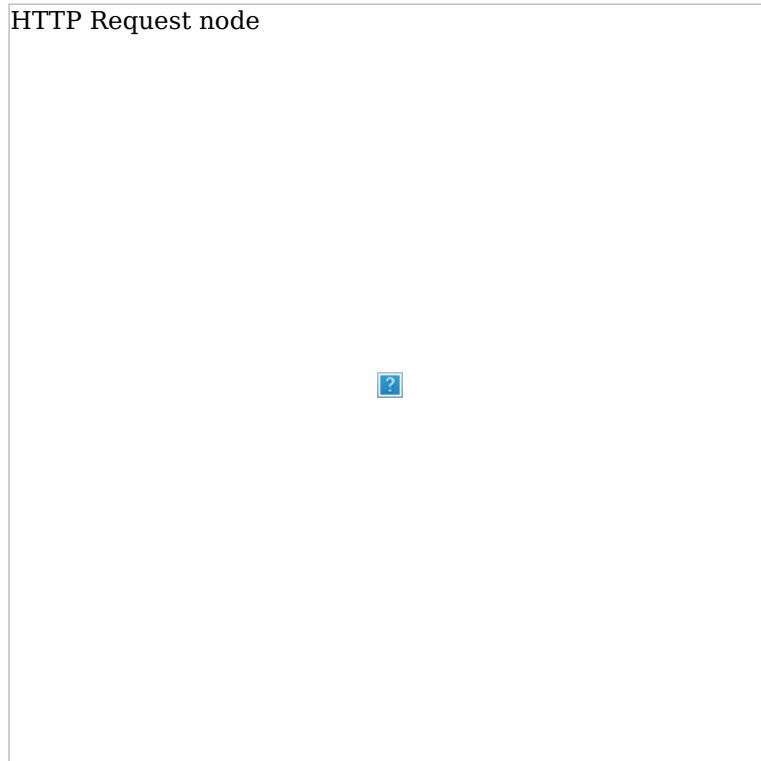
The first thing you need to do is get data from ABCorp's old data warehouse.

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

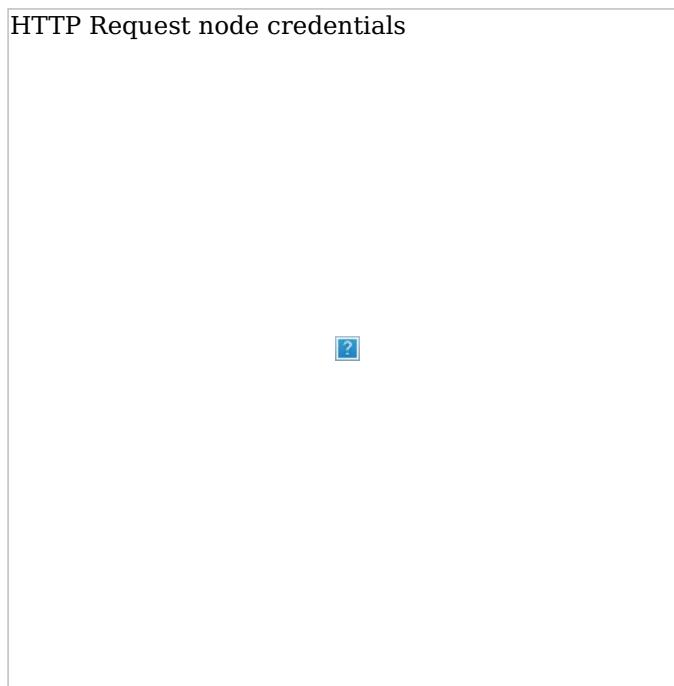
This node will use credentials.

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

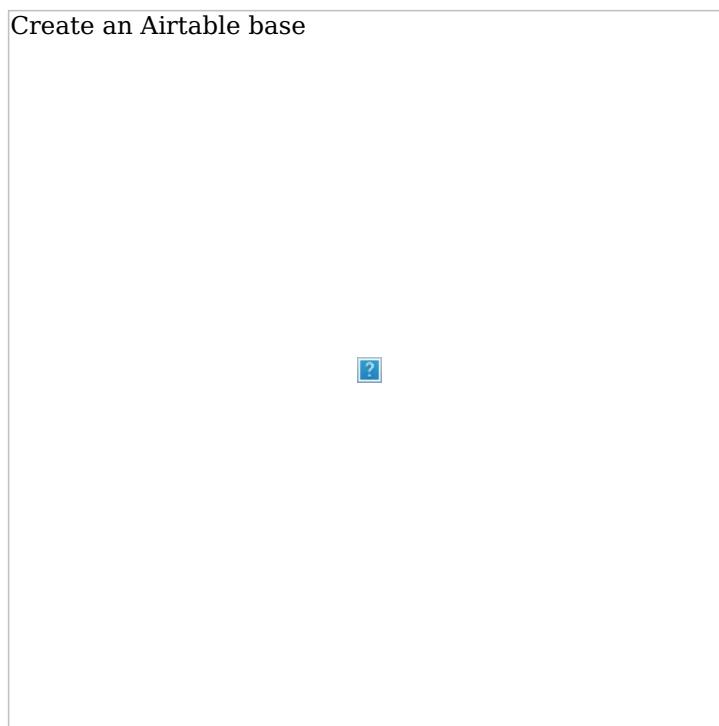
In this step of the workflow, you will learn how to insert the data received from the HTTP Request node into Airtable using the [Airtable node](#).

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*.

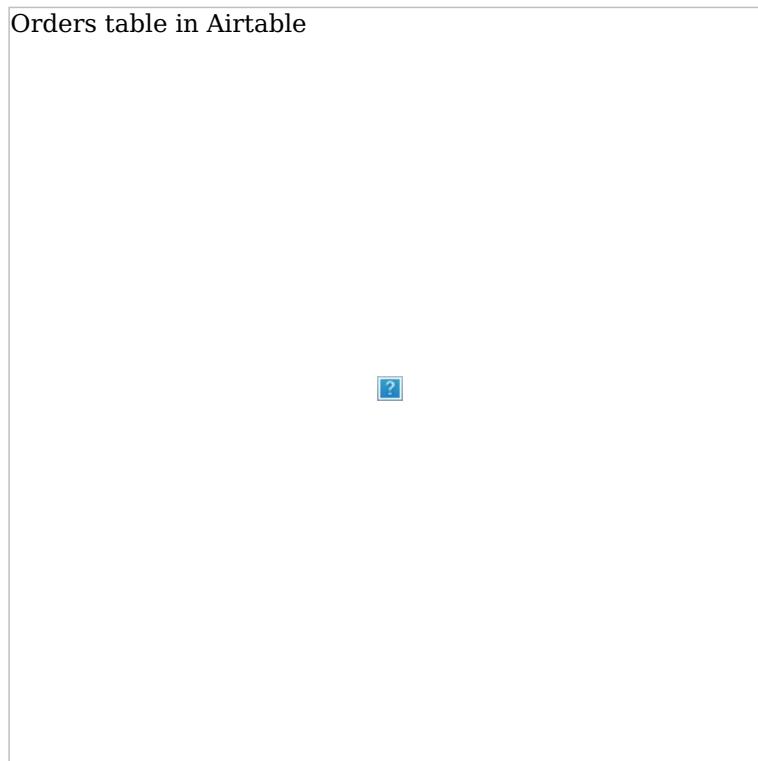


Create an Airtable base

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?

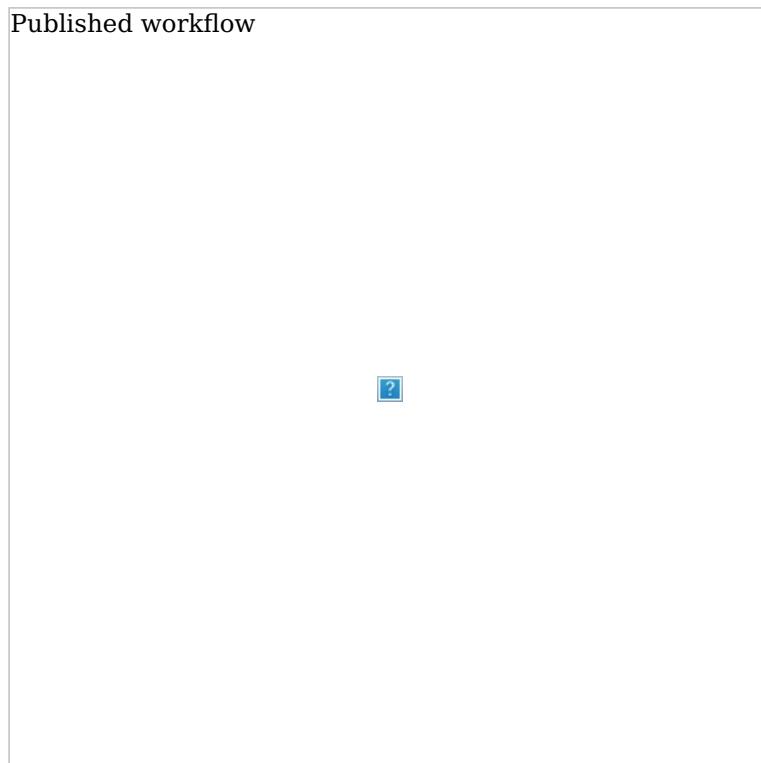
You 🤷: Not so fast. To do that, you need to publish your workflow. I'll do this in the next step and show you how to interpret the execution log.

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.

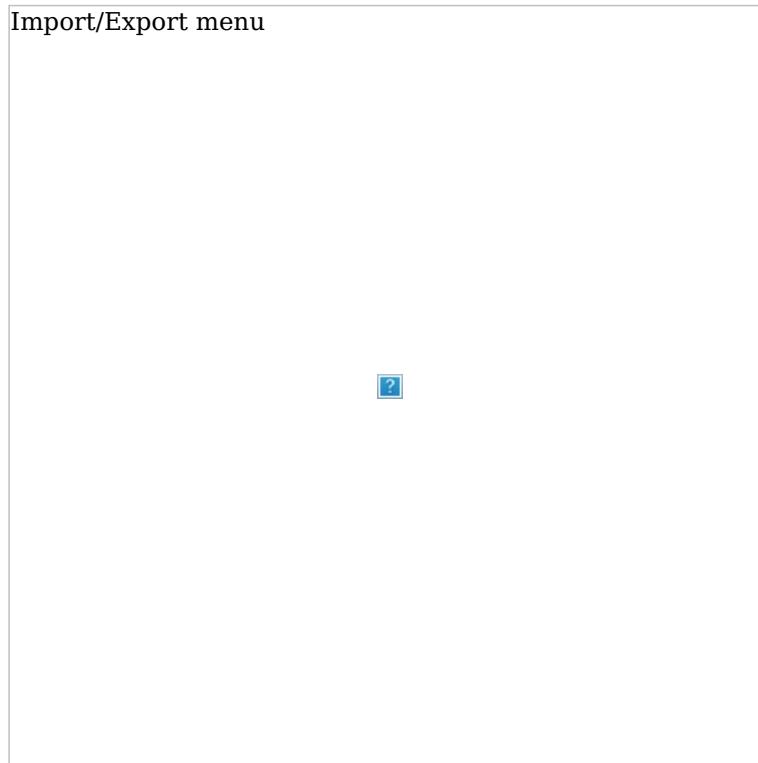
Exporting and importing workflows

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

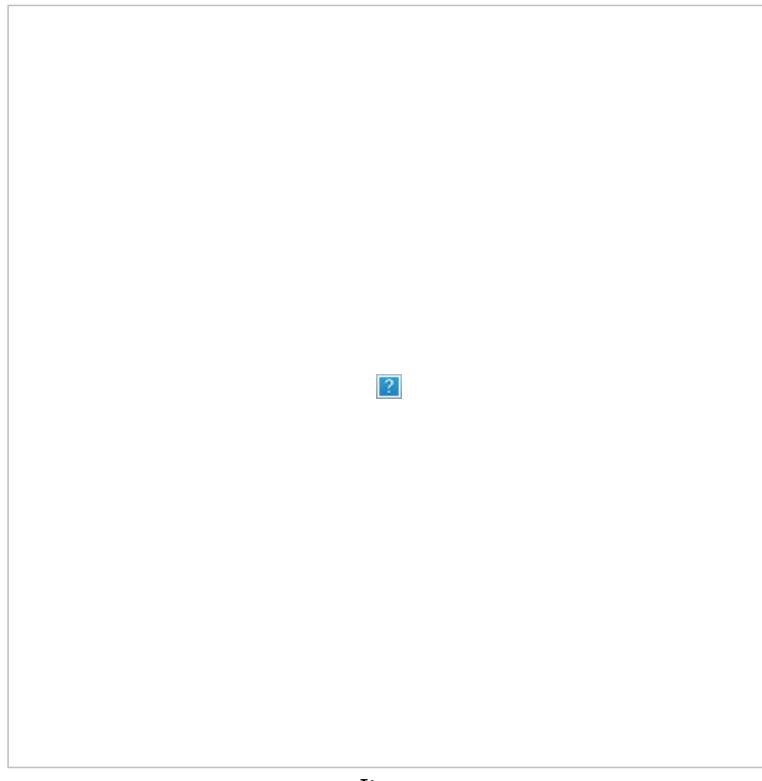
In this chapter, you will learn about the data structure of n8n and how to use the [Code node](#) to transform data and simulate node outputs.

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

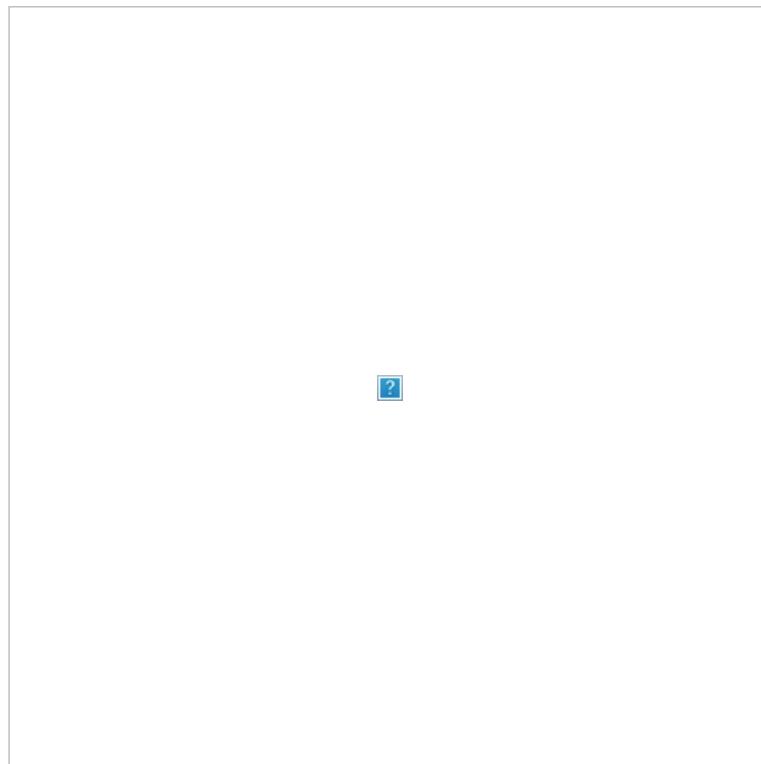
Creating data sets with the Code node

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,
```

```
 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

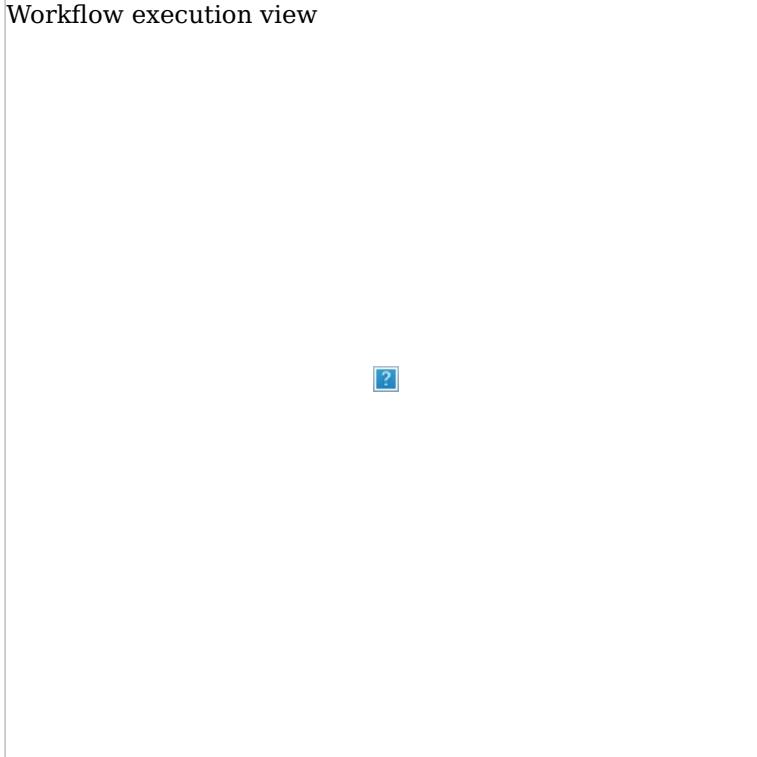


### *Executions log*

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.

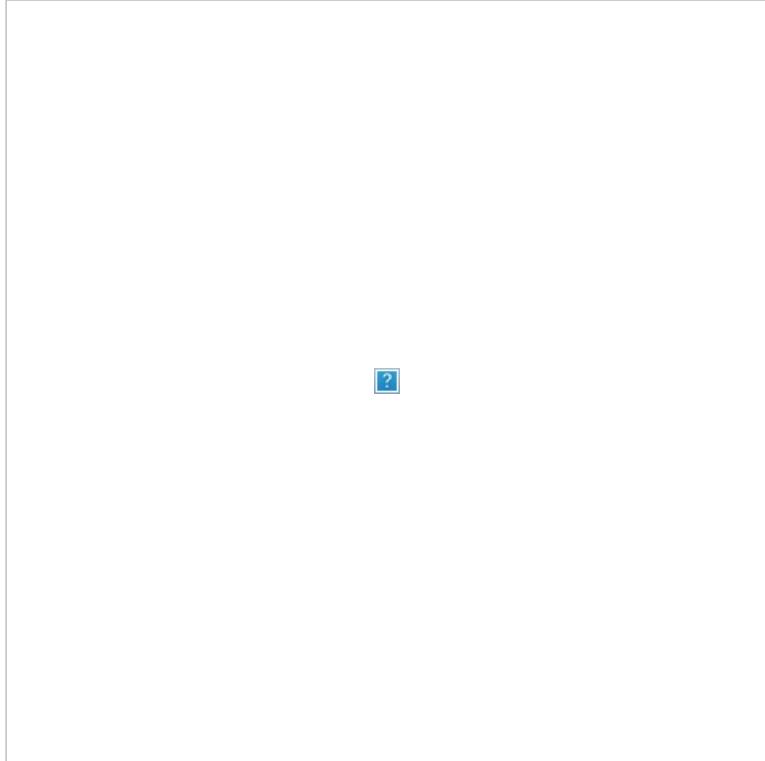
To accomplish this task, you first need to make a copy of this table in your Airtable account:

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



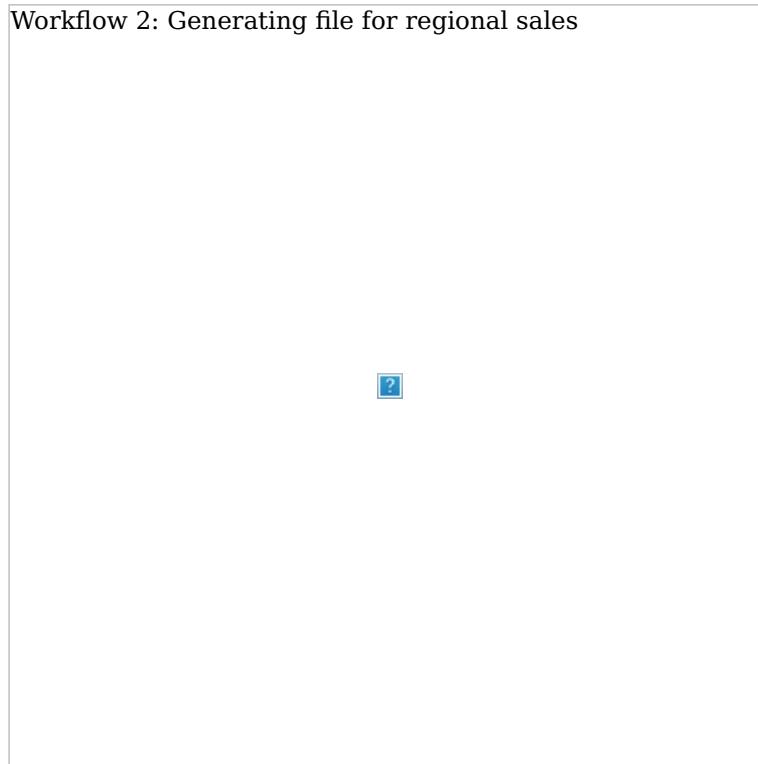
Workflow 1: Getting data from different sources

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

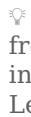
The **Evaluations tab** enables you to review test results in the n8n UI, perfect for comparing runs, spotting regressions, and viewing performance over time.

### ⌘ 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](#)
- [Imperova 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.

# Evaluation Trigger node

Use the Evaluation Trigger node when setting up evaluations to validate your AI workflow reliability. During evaluation, the Evaluation Trigger node reads your evaluation dataset from Google Sheets, sending the items through the workflow one at a time, in sequence.

On this page, you'll find the Evaluation Trigger node parameters and options.

## 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.
  - **Limit Rows:** Whether to limit the number of rows in the data table to process. Default state is off.
    - **Max Rows to Process:** When **Limit Rows** is enabled, the maximum number of rows to read and process during the evaluation. Default value is 10.
    - **Filter Rows:** Whether to filter rows in the data table to process. Default state is off.
- When **Source** is **Google Sheets**:
  - **Credential to connect with:** Create or select an existing [Google Sheets credentials](#).
  - **Document Containing Dataset:** Choose the spreadsheet document with the sheet containing your test dataset.
    - 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 containing your test dataset.
    - 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).
  - **Limit Rows:** Whether to limit the number of rows in the sheet to process.
    - **Max Rows to Process:** When **Limit Rows** is enabled, the maximum number of rows to read and process during the evaluation.
  - **Filters:** Filter the evaluation dataset based on column values.
    - **Column:** Choose a sheet column you want to filter by. Select **From list** to choose the column name from the dropdown list, or **By ID** to specify an ID using an [expression](#).

- **Value:** The column value you want to filter by. The evaluation will only process rows with the given value for the selected column.

## Templates and examples

### Related resources

To learn more about n8n evaluations, check out the [evaluations documentation](#)

n8n provides an app 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.

---

## Execute Command

The Execute Command node runs shell commands on the host machine that runs n8n.

### Node parameters

Configure the node using the following parameters.

#### Execute Once

Choose whether you want the node to execute only once (turned on) or once for every item it receives as input (turned off).

#### Command

Enter the command to execute on the host machine. Refer to sections below for examples of running [multiple commands](#) and [cURL commands](#).

#### Run multiple commands

Use one of two methods to run multiple commands in one Execute Command node:

- Enter each command on one line separated by `&&`. For example, you can combine the change directory (`cd`) command with the list (`ls`) command using `&&`.

```
cd bin && ls
```

- Enter each command on a separate line. For example, you can write the list (`ls`) command on a new line after the change directory (`cd`) command.

```
cd bin
```

[ls](#)

## Run cURL command

You can also use the [HTTP Request](#) node to make a cURL request.

If you want to run the curl command in the Execute Command node, you will have to build a Docker image based on the existing n8n image. The default n8n Docker image uses Alpine Linux. You will have to install the curl package.

1. Create a file named `Dockerfile`.
2. Add the below code snippet to the `Dockerfile`.

```
FROM docker.n8n.io/n8nio/n8n
USER root
RUN apk --update add curl
USER node
```
3. In the same folder, execute the command below to build the Docker image.  
`docker build -t n8n-curl`
4. Replace the Docker image you used before. For example, replace `docker.n8n.io/n8nio/n8n` with `n8n-curl`.
5. Run the newly created Docker image. You'll now be able to execute ssh using the Execute Command Node.

## Templates and examples

## Common issues

For common questions or issues and suggested solutions, refer to [Common Issues](#).

---

## Execute Command node common issues

Here are some common errors and issues with the [Execute Command node](#) and steps to resolve or troubleshoot them.

### Command failed: <command> /bin/sh: <command>: not found

This error occurs when the shell environment can't find one of the commands in the **Command** parameter.

To fix this error, review the following:

- Check that the command and its arguments don't have typos in the

### **Command** parameter.

- Check that the command is in the PATH of the user running n8n.
- If you are running n8n with Docker, check if the command is available within the container by trying to run it manually. If your command isn't included in the container, you might have to extend the official n8n image with a [custom image](#) that includes your command.
  - If n8n is already running: sh # Find n8n's container ID, it will be the first column docker ps | grep n8n # Try to execute the command within the running container docker container exec <container\_ID> <command\_to\_run>
  - If n8n isn't running: sh # Start up a new container that runs the command instead of n8n # Use the same image and tag that you use to run n8n normally docker run -it --rm --entrypoint /bin/sh docker.n8n.io/n8nio/n8n -c <command\_to\_run>

## Error: stdout maxBuffer length exceeded

This error happens when your command returns more output than the Execute Command node is able to process at one time.

To avoid this error, reduce output your command produces. Check your command's manual page or documentation to see if there are flags to limit or filter output. If not, you may need to pipe the output to another command to remove unneeded info.

---

## Execute Sub-workflow

Use the Execute Sub-workflow node to run a different workflow on the host machine that runs n8n.

### Node parameters

#### Source

Select where the node should get the sub-workflow's information from:

- **Database:** Select this option to load the workflow from the database by ID. You must also enter either:
  - **From list:** Select the workflow from a list of workflows available to your account.
  - **Workflow ID:** Enter the ID for the workflow. The URL of the workflow contains the ID after /workflow/. For example, if the URL of a workflow is <https://my-n8n-acct.app.n8n.cloud/workflow/abCDE1f6gHiJKL7>, the **Workflow ID** is abCDE1f6gHiJKL7.
- **Local File:** Select this option to load the workflow from a locally saved JSON file. You must also enter:
  - **Workflow Path:** Enter the path to the local JSON workflow file you want the node to execute.
- **Parameter:** Select this option to load the workflow from a parameter. You must also enter:

- **Workflow JSON:** Enter the JSON code you want the node to execute.
- **URL:** Select this option to load the workflow from a URL. You must also enter:
  - **Workflow URL:** Enter the URL you want to load the workflow from.

## Workflow Inputs

If you select a sub-workflow using the **database** and **From list** options, the sub-workflow's input items will automatically display, ready for you to fill in or map values.

You can optionally remove requested input items, in which case the sub-workflow receives `null` as the item's value. You can also enable **Attempt to convert types** to try to automatically convert data to the sub-workflow item's requested type.

Input items won't appear if the sub-workflow's Workflow Input Trigger node uses the "Accept all data" input data mode.

## Mode

Use this parameter to control the execution mode for the node. Choose from these options:

- **Run once with all items:** Pass all input items into a single execution of the node.
- **Run once for each item:** Execute the node once for each input item in turn.

## Node options

This node includes one option: **Wait for Sub-Workflow Completion**. This lets you control whether the main workflow should wait for the sub-workflow's completion before moving on to the next step (turned on) or whether the main workflow should continue without waiting (turned off).

## Templates and examples

### 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”

---

# Execute Sub-workflow Trigger node

Use this node to start a workflow in response to another workflow. It should be the first node in the workflow.

n8n allows you to call workflows from other workflows. This is useful if you want to:

- Reuse a workflow: for example, you could have multiple workflows pulling and processing data from different sources, then have all those workflows call a single workflow that generates a report.
- Break large workflows into smaller components.

## Usage

This node runs in response to a call from the [Execute Sub-workflow](#) or [Call n8n Workflow Tool](#) nodes.

```
-8<- "_snippets/flow-logic/subworkflow-usage.md"
```

## Templates and examples

## How data passes between workflows

```
-8<- "_snippets/flow-logic/subworkflow-data-flow.md"
```

---

## Execution Data

Use this node to save metadata for workflow executions. You can then search by this data in the **Executions** list.

You can retrieve custom execution data during workflow execution using the Code node. Refer to [Custom executions data](#) for more information.

```
-8<- "_snippets/workflows/executions/custom-execution-data-availability.md"
```

## Operations

- Save Execution Data for Search

## Data to Save

Add a **Saved Field** for each key/value pair of metadata you'd like to save.

## Limitations

The Execution Data node has the following restrictions when storing execution metadata:

- key: limited to 50 characters
- value: limited to 512 characters

If either the key or value exceed the above limitations, n8n truncates to their maximum length and outputs a log entry.

## Templates and examples

---

## Extract From File

A common pattern in n8n workflows is to receive a file, either from an [HTTP Request node](#) (for files you are fetching from a website), a [Webhook Node](#) (for files which are sent to your workflow from elsewhere), or from a local source. Data obtained in this way is often in a binary format, for example a spreadsheet or PDF.

The Extract From File node extracts data from a binary format file and converts it to JSON, which can then be easily manipulated by the rest of your workflow. For converting JSON back into a binary file type, please see the [Convert to File node](#).

## Operations

Use the **Operations** drop-down to select the format of the source file to extract data from.

- **Extract From CSV:** The “Comma Separated Values” file type is commonly used for tabulated data.
- **Extract From HTML:** Extract fields from standard web page HTML format files.
- **Extract From JSON:** Extract JSON data from a binary file.
- **Extract From ICS:** Extract fields from iCalendar format files.
- **Extract From ODS:** Extract fields from ODS spreadsheet files.
- **Extract From PDF:** Extract fields from Portable Document Format files.
- **Extract From RTF:** Extract fields from Rich Text Format files.
- **Extract From Text File:** Extract fields from a standard text file format.
- **Extract From XLS:** Extract fields from a Microsoft Excel file (older format).
- **Extract From XLSX:** Extract fields from a Microsoft Excel file.
- **Move File to Base64 String:** Converts binary data to a text-friendly [base64](#) format.

## Example workflow

In this example, a Webhook node is used to trigger the workflow. When a CSV file is sent to the webhook address, the file data is output and received by the Extract From File node.

Set to operate as 'Extract from CSV', the node then outputs the data as a series of JSON 'row' objects:

```
{
 "row": {
 "0": "apple",
 "1": "1",
 "2": "2",
 "3": "3"
 }
 ...
```

## Node parameters

### Input Binary Field

Enter the name of the field from the node input data that contains the binary file. The default is 'data'.

### Destination Output Field

Enter the name of the field in the node output that will contain the extracted data.

This parameter is only available for these operations:

- Extract From JSON
- Extract From ICS
- Extract From Text File
- Move File to Base64 String

## Templates and examples

---

## Filter

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.

## Node parameters

Create filter comparison **Conditions** to perform your filter.

- Use the data type dropdown to select the data type and comparison operation type for your condition. For example, to filter for dates after a particular date, select **Date & Time > is after**.
- The fields and values to enter into the condition change based on

the data type and comparison you select. Refer to [Available data type comparisons](#) for a full list of all comparisons by data type.

Select **Add condition** to create more conditions.

## Combining conditions

You can choose to keep items:

- When they meet all conditions: Create two or more conditions and select **AND** in the dropdown between them.
- When they meet any of the conditions: Create two or more conditions and select **OR** in the dropdown between them.

You can't create a mix of AND and OR rules.

## Node options

- **Ignore Case:** Whether to ignore letter case (turned on) or be case sensitive (turned off).
- **Less Strict Type Validation:** Whether you want n8n to attempt to convert value types based on the operator you choose (turned on) or not (turned off). Turn this on when facing a "wrong type:" error in your node.

## Templates and examples

-8<- “\_snippets/integrations/builtin/core-nodes/data-types.md”

---

## FTP

The FTP node is useful to access and upload files to an FTP or SFTP server.

To connect to an SFTP server, use an SFTP credential. Refer to [FTP credentials](#) for more information.

## Operations

- **Delete** a file or folder
- **Download** a file
- **List** folder content
- **Rename** or move a file or folder
- **Upload** a file

## Delete

This operation includes one parameter: **Path**. Enter the remote path that you would like to connect to.

## Delete options

The delete operation adds one new option: **Folder**. If you turn this option on, the node can delete both folders and files. This configuration also displays one more option:

- **Recursive**: If you turn this option on and you delete a folder or directory, the node will delete all files and directories within the target directory.

## Download

Configure this operation with these parameters:

- **Path**: Enter the remote path that you would like to connect to.
- **Put Output File in Field**: Enter the name of the output binary field to put the file in.

/// Note | Concurrent Reads with SFTP When using SFTP, you can enable concurrent reads. This improves download speeds but may not be supported by all SFTP servers. ///

## List

Configure this operation with these parameters:

- **Path**: Enter the remote path that you would like to connect to.
- **Recursive**: Select whether to return an object representing all directories / objects recursively found within the FTP/SFTP server (turned on) or not (turned off).

## Rename

Configure this operation with these parameters:

- **Old Path**: Enter the existing path of the file you'd like to rename in this field.
- **New Path**: Enter the new path for the renamed file in this field.

### Rename options

This operation adds one new option: **Create Directories**. If you turn this option on, the node will recursively create the destination directory when renaming an existing file or folder.

## Upload

Configure this operation with these parameters:

- **Path**: Enter the remote path that you would like to connect to.
- **Binary File**: Select whether you'll upload a binary file (turned on) or enter text content to be uploaded (turned off). Other parameters depend on your selection in this field.
  - **Input Binary Field**: Displayed if you turn on **Binary File**. Enter the name of the input binary field that contains the file you'll upload in this field.
  - **File Content**: Displayed if you turn off **Binary File** Enter the

text content of the file you'll upload in this field.

## Templates and examples

---

# Git

Git is a free and open-source distributed version control system designed to handle everything from small to large projects with speed and efficiency.

## Operations

- Add a file or folder to commit. Performs a `git add`.
- Add Config: Add configuration property. Performs a `git config` set or add.
- Clone a repository: Performs a `git clone`.
- Commit files or folders to git. Performs a `git commit`.
- Fetch from remote repository. Performs a `git fetch`.
- List Config: Return current configuration. Performs a `git config` query.
- Log: Return git commit history. Performs a `git log`.
- Pull from remote repository: Performs a `git pull`.
- Push to remote repository: Performs a `git push`.
- Push Tags to remote repository: Performs a `git push -tags`.
- Return Status of current repository: Performs a `git status`.
- Create a new Tag: Performs a `git tag`.
- User Setup: Set the user.

Refer to the sections below for more details on the parameters and options for each operation.

## Add

Configure this operation with these parameters:

- **Repository Path**: Enter the local path of the git repository.
- **Paths to Add**: Enter a comma-separated list of paths of files or folders to add in this field. You can use absolute paths or relative paths from the **Repository Path**.

## Add Config

Configure this operation with these parameters:

- **Repository Path**: Enter the local path of the git repository.
- **Key**: Enter the name of the key to set.
- **Value**: Enter the value of the key to set.

## Add Config options

The add config operation adds the **Mode** option. Choose whether to **Set** or **Append** the setting in the local config.

## Clone

Configure this operation with these parameters:

- **Repository Path:** Enter the local path of the git repository.
- **Authentication:** Select **Authenticate** to pass credentials in. Select **None** to not use authentication.
  - **Credential for Git:** If you select **Authenticate**, you must select or create credentials for the node to use. Refer to [Git credential](#) for more information.
- **New Repository Path:** Enter the local path where you'd like to locate the cloned repository.
- **Source Repository:** Enter the URL or path of the repository you want to clone.

## Commit

Configure this operation with these parameters:

- **Repository Path:** Enter the local path of the git repository.
- **Message:** Enter the commit message to use in this field.

### Commit options

The commit operation adds the **Paths to Add** option. To commit all “added” files and folders, leave this field blank. To commit specific “added” files and folders, enter a comma-separated list of paths of files or folders in this field.

You can use absolute paths or relative paths from the **Repository Path**.

## Fetch

This operation only prompts you to enter the local path of the git repository in the **Repository Path** parameter.

## List Config

This operation only prompts you to enter the local path of the git repository in the **Repository Path** parameter.

## Log

Configure this operation with these parameters:

- **Repository Path:** Enter the local path of the git repository.
- **Return All:** When turned on, the node will return all results. When turned off, the node will return results up to the set **Limit**.
- **Limit:** Only available when you turn off **Return All**. Enter the

maximum number of results to return.

## Log options

The log operation adds the **File** option. Enter the path of a file or folder to get the history of in this field.

You can use absolute paths or relative paths from the **Repository Path**.

## Pull

This operation only prompts you to enter the local path of the git repository in the **Repository Path** parameter.

## Push

Configure this operation with these parameters:

- **Repository Path:** Enter the local path of the git repository.
- **Authentication:** Select **Authenticate** to pass credentials in or **None** to not use authentication.
  - If you select **Authenticate**, you must select or create **Credential for Git** for the node to use. Refer to [Git credential](#) for more information.

## Push options

The push operation adds the **Target Repository** option. Enter the URL or path of the repository to push to in this field.

## Push Tags

This operation only prompts you to enter the local path of the git repository in the **Repository Path** parameter.

## Status

This operation only prompts you to enter the local path of the git repository in the **Repository Path** parameter.

## Tag

Configure this operation with these parameters:

- **Repository Path:** Enter the local path of the git repository.
- **Name:** Enter the name of the tag to create in this field.

## User Setup

This operation only prompts you to enter the local path of the git repository in the **Repository Path** parameter.

## Templates and examples

---

## GraphQL

GraphQL is an open-source data query and manipulation language for APIs, and a runtime for fulfilling queries with existing data. Use the GraphQL node to query a GraphQL endpoint.

### Node parameters

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

#### Authentication

Select the type of authentication to use.

If you select anything other than **None**, the **Credential for** parameter appears for you to select an existing or create a new authentication credential for that authentication type.

#### HTTP Request Method

Select the underlying HTTP Request method the node should use.  
Choose from:

- **GET**
- **POST**: If you select this method, you'll also need to select the **Request Format** the node should use for the query payload.  
Choose from:
  - **GraphQL (Raw)**
  - **JSON**

#### Endpoint

Enter the GraphQL Endpoint you'd like to hit.

#### Ignore SSL Issues

When you turn on this control, n8n ignores SSL certificate validation failure.

#### Query

Enter the GraphQL query you want to execute.

Refer to [Related Resources](#) for information on writing your query.

#### Response Format

Select the format you'd like to receive query results in. Choose between:

- **JSON**
- **String**: If you select this format, enter a **Response Data Property Name** to define the property the string is written to.

## Headers

Enter any **Headers** you want to pass as part of the query as **Name / Value** pairs.

## Templates and examples

## Related resources

To use the GraphQL node, you need to understand GraphQL query language. GraphQL have their own [Introduction to GraphQL](#) tutorial.

---

## Guardrails node

Use the Guardrails node to enforce safety, security, and content policies on text. You can use it to validate user input *before* sending it to an AI model, or to check the *output* from an AI model before using it in your workflow.

### Node parameters

Use these parameters to configure the Guardrails node.

### Operation

The operation mode for this node to define its behavior.

- **Check Text for Violations**: Provides a full set of guardrails. Any violation will send items to **Fail** branch.
- **Sanitize Text**: Provides a subset of guardrails that can detect URLs, regular expressions, secret keys, or personally identifiable information (PII), such as phone numbers and credit card numbers. The node replaces detected violations with placeholders.

### Text To Check

The text the guardrails evaluate. Typically, you map this text using an expression from a previous node, such as text from a user query or a response from an AI model.

### Guardrails

Select one or more guardrails to apply to the **Text To Check**. When you add a guardrail from the list, its specific configuration options appear below.

- **Keywords:** Checks if specified keywords appear in the input text.
  - **Keywords:** A comma-separated list of words to block.
- **Jailbreak:** Detects attempts to bypass AI safety measures or exploit the model.
  - **Customize Prompt:** (Boolean) If you turn this on, a text input appears with the default prompt for the jailbreak detection model. You can change this prompt to fine-tune the guardrail.
  - **Threshold:** A value between 0.0 and 1.0. This represents the confidence level required from the AI model to flag the input as a jailbreak attempt. A higher threshold is stricter.
- **NSFW:** Detects attempts to generate Not Safe For Work (NSFW) content.
  - **Customize Prompt:** (Boolean) If you turn this on, a text input appears with the default prompt for the NSFW detection model. You can change this prompt to fine-tune the guardrail.
  - **Threshold:** A value between 0.0 and 1.0 representing the confidence level required to flag the content as NSFW.
- **PII:** Detects personally identifiable information (PII) in the text.
  - **Type:** Choose which PII entities to scan for:
    - **All:** Scans for all available entity types.
    - **Selected:** Allows you to choose specific entities from a list.
  - **Entities:** (Appears if **Type** is **Selected**) A multi-select list of PII types to detect (for example, CREDIT\_CARD, EMAIL\_ADDRESS, PHONE\_NUMBER, and US\_SSN).
- **Secret Keys:** Detects the presence of secret keys or API credentials in the text.
  - **Permissiveness:** How strict or permissive the detection should be when flagging secret keys:
    - **Strict**
    - **Permissive**
    - **Balanced**
- **Topical Alignment:** Ensures the conversation stays within a predefined scope or topic (also known as “business scope”).
  - **Prompt:** A preset prompt that defines the *allowed* topic. The guardrail checks if the **Text To Check** aligns with this prompt.
  - **Threshold:** A value between 0.0 and 1.0 representing the confidence level required to flag the input as *off-topic*.
- **URLs:** Manages URLs the node finds in the input text. It detects all URLs as violations, unless you specify them in **Block All URLs Except**.
  - **Block All URLs Except:** (Optional) A comma-separated list of URLs that you permit.
  - **Allowed Schemes:** Select the URL schemes to permit (for example, https, http, ftp, and mailto).
  - **Block userinfo:** (Boolean) If you turn this on, the node blocks URLs containing user credentials (for example, user:pass@example.com) to prevent credential injection.
  - **Allow subdomain:** (Boolean) If you turn this on, the node automatically allows subdomains of any URL in the **Block All URLs Except** list (for example, sub.example.com would be allowed if example.com is in the list).
- **Custom:** Define your own custom, LLM-based guardrail.
  - **Name:** A descriptive name for your custom guardrail (for example, “Check for rude language”).
  - **Prompt:** A prompt that instructs the AI model what to check

- for.
- **Threshold:** A value between 0.0 and 1.0 representing the confidence level required to flag the input as a violation.
  - **Custom Regex:** Define your own custom regular expression patterns.
    - **Name:** A name for your custom pattern. The node uses this name as a placeholder in the **Sanitize Text** mode.
    - **Regex:** Your regular expression pattern.

## Customize System Message

If you turn this on, a text input appears with a message that the guardrail uses to enforce thresholds and JSON output according to schema. Change it to modify the global guardrails behavior.

---

# HTML

The HTML node provides operations to help you work with HTML in n8n.

## Operations

- **Generate HTML template:** Use this operation to create an HTML template. This allows you to take data from your workflow and output it as HTML.
- **Extract HTML content:** Extract contents from an HTML-formatted source. The source can be in JSON or a binary file (.html).
- **Convert to HTML Table:** Convert content to an HTML table.

The node parameters and options depend on the operation you select. Refer to the sections below for more details on configuring each operation.

## Generate HTML template

Create an HTML template. This allows you to take data from your workflow and output it as HTML.

You can include:

- Standard HTML
- CSS in `<style>` tags.
- JavaScript in `<script>` tags. n8n doesn't execute the JavaScript.
- Expressions, wrapped in `{()}`.

You can use Expressions in the template, including n8n's Built-in methods and variables.

## Extract HTML Content

Extract contents from an HTML-formatted source. The source can be in JSON or a binary file (.html).

Use these parameters:

## Source Data

Select the source type for your HTML content. Choose between:

- **JSON**: If you select this source data, enter the **JSON Property**: the name of the input containing the HTML you want to extract. The property can contain a string or an array of strings.
- **Binary**: If you select this source data, enter the **Input Binary Field**: the name of the input containing the HTML you want to extract. The property can contain a string or an array of strings.

## Extraction Values

- **Key**: Enter the key to save the extracted value under.
- **CSS Selector**: Enter the CSS selector to search for.
- **Return Value**: Select the type of data to return. Choose from:
  - **Attribute**: Return an attribute value like class from an element.
    - If you select this option, enter the name of the **Attribute** to return the value of.
  - **HTML**: Return the HTML that the element contains.
  - **Text**: Return the text content of the element.
    - If you choose this option, you can also enter a comma-separated list of selectors to skip in the **Skip Selectors**.
  - **Value**: Return the value of an input, select, or text area.
- **Return Array**: Choose whether to return multiple extraction values as an array (turned on) or as a single string (turned off).

## Extract HTML Content options

You can also configure this operation with these options:

- **Trim Values**: Controls whether to remove all spaces and newlines from the beginning and end of the values (turned on) or leaves them (turned off).
- **Clean Up Text**: Controls whether to remove leading whitespaces, trailing whitespaces, and line breaks (newlines) and condense multiple consecutive whitespaces into a single space (turned on) or to leave them as-is (turned off).

## Convert to HTML Table

This operation expects data from another node. It has no parameters. It includes these options:

- **Capitalize Headers**: Controls whether to capitalize the table's headers (turned on) or not (turned off).
- **Custom Styling**: Controls whether to use custom styling (turned on) or not (turned off).
- **Caption**: Enter a caption to add to the table.
- **Table Attributes**: Enter any attributes to apply to the <table>, such as style attributes.
- **Header Attributes**: Enter any attributes to apply to the table's headers <th>.
- **Row Attributes**: Enter any attributes to apply to the table's rows

- <tr>.  
• **Cell Attributes:** Enter any attributes to apply to the table's cells  
<td>.

## Templates and examples

---

## HTTP Request node

The HTTP Request node is one of the most versatile nodes in n8n. It allows you to make HTTP requests to query data from any app or service with a REST API. You can use the HTTP Request node a regular node or attached to an [AI agent](#) to use as a `tool{ data-preview }`.

When using this node, you're creating a REST API call. You need some understanding of basic API terminology and concepts.

There are two ways to create an HTTP request: configure the [node parameters](#) or [import a curl command](#).

## Node parameters

### Method

Select the method to use for the request:

- DELETE
- GET
- HEAD
- OPTIONS
- PATCH
- POST
- PUT

### URL

Enter the endpoint you want to use.

### Authentication

n8n recommends using the **Predefined Credential Type** option when it's available. It offers an easier way to set up and manage credentials, compared to configuring generic credentials.

### Predefined credentials

Credentials for integrations supported by n8n, including both built-in and community nodes. Use **Predefined Credential Type** for custom operations without extra setup. Refer to [Custom API operations](#) for more information.

### Generic credentials

Credentials for integrations not supported by n8n. You'll need to manually configure the authentication process, including specifying the required API endpoints, necessary parameters, and the authentication method.

You can select one of the following methods:

- Basic auth
- Custom auth
- Digest auth
- Header auth
- OAuth1 API
- OAuth2 API
- Query auth

Refer to [HTTP request credentials](#) for more information on setting up each credential type.

## Send Query Parameters

Query parameters act as filters on HTTP requests. If the API you're interacting with supports them and the request you're making needs a filter, turn this option on.

**Specify your query parameters** using one of the available options:

- **Using Fields Below:** Enter **Name/Value** pairs of **Query Parameters**. To enter more query parameter name/value pairs, select **Add Parameter**. The name is the name of the field you're filtering on, and the value is the filter value.
- **Using JSON:** Enter **JSON** to define your query parameters.

Refer to your service's API documentation for detailed guidance.

## Send Headers

Use this parameter to send headers with your request. Headers contain metadata or context about your request.

**Specify Headers** using one of the available options:

- **Using Fields Below:** Enter **Name/Value** pairs of **Header Parameters**. To enter more header parameter name/value pairs, select **Add Parameter**. The name is the header you wish to set, and the value is the value you want to pass for that header.
- **Using JSON:** Enter **JSON** to define your header parameters.

Refer to your service's API documentation for detailed guidance.

## Send Body

If you need to send a body with your API request, turn this option on.

Then select the **Body Content Type** that best matches the format for the body content you wish to send.

### Form URLEncoded

Use this option to send your body as application/x-www-form-urlencoded.

**Specify Body** using one of the available options:

- **Using Fields Below:** Enter **Name/Value** pairs of **Body Parameters**. To enter more body parameter name/value pairs, select **Add Parameter**. The name should be the form field name, and the value is what you wish to set that field to.
- **Using Single Field:** Enter your name/value pairs in a single **Body** parameter with format `fieldname1=value1&fieldname2=value2`.

Refer to your service's API documentation for detailed guidance.

## Form-Data

Use this option to send your body as multipart/form-data.

Configure your **Body Parameters** by selecting the **Parameter Type**:

- Choose **Form Data** to enter **Name/Value** pairs.
- Choose **n8n Binary File** to pull the body from a file the node has access to.
  - **Name:** Enter the ID of the field to set.
  - **Input Data Field Name:** Enter the name of the incoming field containing the binary file data you want to process.

Select **Add Parameter** to enter more parameters.

Refer to your service's API documentation for detailed guidance.

## JSON

Use this option to send your body as JSON.

**Specify Body** using one of the available options:

- **Using Fields Below:** Enter **Name/Value** pairs of **Body Parameters**. To enter more body parameter name/value pairs, select **Add Parameter**.
- **Using JSON:** Enter **JSON** to define your body.

Refer to your service's API documentation for detailed guidance.

## n8n Binary File

Use this option to send the contents of a file stored in n8n as the body.

Enter the name of the incoming field that contains the file as the **Input Data Field Name**.

Refer to your service's API documentation for detailed guidance on how to format the file.

## Raw

Use this option to send raw data in the body.

- **Content Type:** Enter the Content-Type header to use for the raw body content. Refer to the IANA [Media types](#) documentation for a

full list of MIME content types.

- **Body:** Enter the raw body content to send.

Refer to your service's API documentation for detailed guidance.

## Node options

Select **Add Option** to view and select these options. Options are available to all parameters unless otherwise noted.

### Array Format in Query Parameters

Use this option to control the format for arrays included in query parameters. Choose from these options:

- **No Brackets:** Arrays will format as the name=value for each item in the array, for example: foo=bar&foo=qux.
- **Brackets Only:** The node adds square brackets after each array name, for example: foo[] = bar & foo[] = qux.
- **Brackets with Indices:** The node adds square brackets with an index value after each array name, for example: foo[0]=bar&foo[1]=qux.

Refer to your service's API documentation for guidance on which option to use.

### Batching

Control how to batch large numbers of input items:

- **Items per Batch:** Enter the number of input items to include in each batch.
- **Batch Interval:** Enter the time to wait between each batch of requests in milliseconds. Enter 0 for no batch interval.

### Ignore SSL Issues

By default, n8n only downloads the response if SSL certificate validation succeeds. If you'd like to download the response even if SSL certificate validation fails, turn this option on.

### Lowercase Headers

Choose whether to lowercase header names (turned on, default) or not (turned off).

### Redirects

Choose whether to follow redirects (turned on by default) or not (turned off). If turned on, enter the maximum number of redirects the request should follow in **Max Redirects**.

### Response

Use this option to set some details about the expected API response, including:

- **Include Response Headers and Status:** By default, the node returns only the body. Turn this option on to return the full response (headers and response status code) as well as the body.
- **Never Error:** By default, the node returns success only when the response returns with a 2xx code. Turn this option on to return success regardless of the code returned.
- **Response Format:** Select the format in which the data gets returned. Choose from:
  - **Autodetect** (default): The node detects and formats the response based on the data returned.
  - **File:** Select this option to put the response into a file. Enter the field name where you want the file returned in **Put Output in Field**.
  - **JSON:** Select this option to format the response as JSON.
  - **Text:** Select this option to format the response as plain text. Enter the field name where you want the file returned in **Put Output in Field**.

## Pagination

Use this option to paginate results, useful for handling query results that are too big for the API to return in a single call.

??? Details “Understand pagination” Pagination means splitting a large set of data into multiple pages. The amount of data on each page depends on the limit you set.

For example, you make an API call to an endpoint called `/users`. The API wants to send back information on 300 users, but this is too much data for the API to send in one response.

If the API supports pagination, you can incrementally fetch the data. To do this, you call `/users` with a pagination limit, and a page number or URL to tell the API which page to send. In this example, say you use a limit of 10, and start from page 0. The API sends the first 10 users in its response. You then call the API again, increasing the page number by 1, to get the next 10 results.

Configure the pagination settings:

- **Pagination Mode:**
  - **Off:** Turn off pagination.
  - **Update a Parameter in Each Request:** Use this when you need to dynamically set parameters for each request.
  - **Response Contains Next URL:** Use this when the API response includes the URL of the next page. Use an expression to set **Next URL**.

For example setups, refer to [HTTP Request node cookbook | Pagination](#).

n8n provides built-in variables for working with HTTP node requests and responses when using pagination:

```
-8<- "snippets/integrations/builtin/core-nodes/http/pagination-variables.md"
```

```
-8<- "snippets/integrations/builtin/core-nodes/http/pagination-api-differences.md"
```

## Proxy

Use this option if you need to specify an HTTP proxy.

Enter the **Proxy** the request should use. This takes precedence over global settings defined with the [HTTP\\_PROXY, HTTPS\\_PROXY, or ALL\\_PROXY environment variables](#).

## Timeout

Use this option to set how long the node should wait for the server to send response headers (and start the response body). The node aborts requests that exceed this value for the initial response.

Enter the **Timeout** time to wait in milliseconds.

## Tool-only options

The following options are only available when attached to an [AI agent](#) as a `tool{ data-preview }`.

### Optimize Response

Whether to optimize the tool response to reduce the amount of data passed to the LLM. Optimizing the response can reduce costs and can help the LLM ignore unimportant details, often leading to better results.

When optimizing responses, you select an expected response type, which determines other options you can configure. The supported response types are:

#### JSON

When expecting a **JSON** response, you can configure which parts of the JSON data to use as a response with the following choices:

- **Field Containing Data:** This field identifies a specific part of the JSON object that contains your relevant data. You can leave this blank to use the entire response.
- **Include Fields:** This is how you choose which fields you want in your response object. There are three choices:
  - **All:** Include all fields in the response object.
  - **Selected:** Include only the fields specified below.
    - **Fields:** A comma-separated list of fields to include in the response. You can use dot notation to specify nested fields. You can drag fields from the Input panel to add them to the field list.
  - **Exclude:** Include all fields *except* the fields specified below.
    - **Fields:** A comma-separated list of fields to exclude from the response. You can use dot notation to specify nested fields. You can drag fields from the Input panel to add them to the field list.

#### HTML

When expecting **HTML**, you can identify the part of an HTML document relevant to the LLM and optimize the response with the following options:

- **Selector (CSS)**: A specific element or element type to include in the response HTML. Uses the body element by default.
- **Return Only Content**: Whether to strip HTML tags and attributes from the response, leaving only the actual content. This uses fewer tokens and may be easier for the model to understand.
  - **Elements To Omit**: A comma-separated list of CSS selectors to exclude when extracting content.
- **Truncate Response**: Whether to limit the response size to save tokens.
  - **Max Response Characters**: The maximum number of characters to include in the HTML response. The default value is 1000.

## Text

When expecting a generic **Text** response, you can optimize the results with the following options:

- **Truncate Response**: Whether to limit the response size to save tokens.
  - **Max Response Characters**: The maximum number of characters to include in the HTML response. The default value is 1000.

## Import curl command

`curl` is a command line tool and library for transferring data with URLs.

You can use curl to call REST APIs. If the API documentation of the service you want to use provides curl examples, you can copy them out of the documentation and into n8n to configure the HTTP Request node.

Import a curl command:

1. From the HTTP Request node's **Parameters** tab, select **Import cURL**. The **Import cURL command** modal opens.
2. Paste your curl command into the text box.
3. Select **Import**. n8n loads the request configuration into the node fields. This overwrites any existing configuration.

## Templates and examples

## Common issues

For common questions or issues and suggested solutions, refer to [Common Issues](#).

---

# HTTP Request node common issues

Here are some common errors and issues with the [HTTP Request node](#) and steps to resolve or troubleshoot them.

## Bad request - please check your parameters

This error displays when the node receives a 400 error indicating a bad request. This error most often occurs because:

- You're using an invalid name or value in a **Query Parameter**.
- You're passing array values in a **Query Parameter** but the array isn't formatted correctly. Try using the [Array Format in Query Parameters](#) option.

Review the API documentation for your service to format your query parameters.

## The resource you are requesting could not be found

This error displays when the endpoint **URL** you entered is invalid.

This may be due to a typo in the URL or a deprecated API. Refer to your service's API documentation to verify you have a valid endpoint.

## JSON parameter need to be an valid JSON

This error displays when you've passed a parameter as JSON and it's not formatted as valid JSON.

To resolve, review the JSON you've entered for these issues:

- Test your JSON in a JSON checker or syntax parser to find errors like missing quotation marks, extra or missing commas, incorrectly formatted arrays, extra or missing square brackets or curly brackets, and so on.
- If you've used an **Expression** in the node, be sure you've wrapped the entire JSON in double curly brackets, for example: {{  
    "myjson": { "name1": "value1",  
               "array1": [ "value1", "value2" ] } }}}

## Forbidden - perhaps check your credentials

This error displays when the node receives a 403 error indicating authentication failed.

To resolve, review the selected credentials and make sure you can authenticate with them. You may need to:

- Update permissions or scopes so that your API key or account can perform the operation you've selected.
- Format your generic credential in a different way.
- Generate a new API key or token with the appropriate permissions or scopes.

## 429 - The service is receiving too many requests from you

This error displays when the node receives a [429 error](#) from the service that you're calling. This often means that you have hit the rate limits of that service. You can find out more on the [Handling API rate limits](#) page.

To resolve the error, you can use one of the built-in options of the HTTP request node:

### Batching

Use this option to send requests in batches and introduce a delay between them.

1. In the HTTP Request node, select **Add Option > Batching**.
2. Set **Items per Batch** to the number of input items to include in each request.
3. Set **Batch Interval (ms)** to introduce a delay between requests in milliseconds. For example, to send one request to an API per second, set **Batch Interval (ms)** to 1000.

### Retry on Fail

Use this option to retry the node after a failed attempt.

1. In the HTTP Request node, go to **Settings** and enable **Retry on Fail**.
  2. Set **Max Tries** to the maximum number of times n8n should retry the node.
  3. Set **Wait Between Tries (ms)** to the desired delay in milliseconds between retries. For example, to wait one second before retrying the request again, set **Wait Between Tries (ms)** to 1000.
- 

## If

Use the If node to split a workflow conditionally based on comparison operations.

### Add conditions

Create comparison **Conditions** for your If node.

- Use the data type dropdown to select the data type and comparison operation type for your condition. For example, to filter for dates after a particular date, select **Date & Time > is after**.
- The fields and values to enter into the condition change based on the data type and comparison you select. Refer to [Available data type comparisons](#) for a full list of all comparisons by data type.

Select **Add condition** to create more conditions.

## Combining conditions

You can choose to keep data:

- When it meets all conditions: Create two or more conditions and select **AND** in the dropdown between them.
- When it meets any of the conditions: Create two or more conditions and select **OR** in the dropdown between them.

## Templates and examples

### Branch execution with If and Merge nodes

-8<- “\_snippets/integrations/builtin/core-nodes/merge/if-merge-branch-execution.md”

## Related resources

Refer to [Splitting with conditionals](#) for more information on using conditionals to create complex logic in n8n.

If you need more than two conditional outputs, use the [Switch node](#).

-8<- “\_snippets/integrations/builtin/core-nodes/data-types.md”

---

## JWT

Work with JSON web tokens in your n8n workflows.

## Operations

- Decode
- Sign
- Verify

## Node parameters

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

- **Credential to connect with:** Select or create a [JWT credential](#) to connect with.

- **Token:** Enter the token to **Verify** or **Decode**.
- If you select the **Sign** operation, you'll also have this parameter:
  - **Use JSON to Build Payload:** When turned on, the node uses JSON to build the claims. The selection here influences what appears in the Payload Claims section.

## Payload Claims

The node only displays payload claims if you select the **Sign** operation. What you see depends on what you select for **Use JSON to Build Payload:**

- If you select **Use JSON to Build Payload**, this section displays a JSON editor where you can construct the claims.
- If you don't select **Use JSON to Build Payload**, this section prompts you to **Add Claim**.

You can add the following claims.

### Audience

The **Audience** or `aud` claim identifies the intended recipients of the JWT.

Refer to [“aud” \(Audience\) Claim](#) for more information.

### Expires In

The **Expires In** or `exp` claim identifies the time after which the JWT expires and must not be accepted for processing.

Refer to [“exp” \(Expiration Time\) Claim](#) for more information.

### Issuer

The **Issuer** or `iss` claim identifies the principal that issued the JWT.

Refer to [“iss” \(Issuer\) Claim](#) for more information.

### JWT ID

The **JWT ID** or `jti` claim provides a unique identifier for the JWT.

Refer to [“jti” \(JWT ID\) Claim](#) for more information.

### Not Before

The **Not Before** or `nbf` claim identifies the time before which the JWT must not be accepted for processing.

Refer to [“nbf” \(Not Before\) Claim](#) for more information.

### Subject

The **Subject** or `sub` claim identifies the principal that's the subject of the JWT.

Refer to “[sub](#)” ([Subject](#)) [Claim](#) for more information.

## Node options

### Decode node options

The **Return Additional Info** toggle controls how much information the node returns.

When turned on, the node returns the complete decoded token with information about the header and signature. When turned off, the node only returns the payload.

### Sign node options

Use the **Override Algorithm** control to select the algorithm to use for verifying the token. This algorithm will override the algorithm selected in the credentials.

### Verify node options

This operation includes several node options:

- **Return Additional Info:** This toggle controls how much information the node returns. When turned on, the node returns the complete decoded token with information about the header and signature. When turned off, the node only returns the payload.
- **Ignore Expiration:** This toggle controls whether the node should ignore the token’s expiration time claim (exp). Refer to “[exp](#)” ([Expiration Time](#)) [Claim](#) for more information.
- **Ignore Not Before Claim:** This toggle controls whether to ignore the token’s not before claim (nbf). Refer to “[nbf](#)” ([Not Before](#)) [Claim](#) for more information.
- **Clock Tolerance:** Enter the number of seconds to tolerate when checking the nbf and exp claims. This allows you to deal with small clock differences among different servers. Refer to “[exp](#)” ([Expiration Time](#)) [Claim](#) for more information.
- **Override Algorithm:** The algorithm to use for verifying the token. This algorithm will override the algorithm selected in the credentials.

## Templates and examples

---

## LDAP

This node allows you to interact with your LDAP servers to create, find, and update objects.

## Operations

- [Compare](#) an attribute

- **Create** a new entry
- **Delete** an entry
- **Rename** the DN of an existing entry
- **Search** LDAP
- **Update** attributes

Refer to the sections below for details on configuring the node for each operation.

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

## Compare

Configure this operation using these parameters:

- **Credential to connect with:** Select or create an [LDAP credential](#) to connect with.
- **DN:** Enter the Distinguished Name (DN) of the entry to compare.
- **Attribute ID:** Enter the ID of the attribute to compare.
- **Value:** Enter the value to compare.

## Create

Configure this operation using these parameters:

- **Credential to connect with:** Select or create an [LDAP credential](#) to connect with.
- **DN:** Enter the Distinguished Name (DN) of the entry to create.
- **Attributes:** Add the **Attribute ID/Value** pairs you'd like to create.

## Delete

Configure this operation using these parameters:

- **Credential to connect with:** Select or create an [LDAP credential](#) to connect with.
- **DN:** Enter the Distinguished Name (DN) of the entry to be deleted.

## Rename

Configure this operation using these parameters:

- **Credential to connect with:** Select or create an [LDAP credential](#) to connect with.
- **DN:** Enter the current Distinguished Name (DN) of the entry to rename.
- **New DN:** Enter the new Distinguished Name (DN) for the entry in this field.

## Search

Configure this operation using these parameters:

- **Credential to connect with:** Select or create an [LDAP credential](#)

to connect with.

- **Base DN:** Enter the Distinguished Name (DN) of the subtree to search in.
- **Search For:** Select the directory object class to search for.
- **Attribute:** Select the attribute to search for.
- **Search Text:** Enter the text to search for. Use \* for a wildcard.
- **Return All:** When turned on, the node will return all results. When turned off, the node will return results up to the set **Limit**.
- **Limit:** Only available when you turn off **Return All**. Enter the maximum number of results to return.

## Search options

You can also configure this operation using these options:

- **Attribute Names or IDs:** Enter a comma-separated list of attributes to return. Choose from the list or specify IDs using an expression.
- **Page Size:** Enter the maximum number of results to request at one time. Set to 0 to disable paging.
- **Scopes:** The set of entries at or below the **Base DN** to search for potential matches. Select from:
  - **Base Tree:** Often referred to as subordinateSubtree or just “subordinates,” selecting this option will search the subordinates of the **Base DN** entry but not the **Base DN** entry itself.
  - **Single Level:** Often referred to as “one,” selecting this option will search only the immediate children of the **Base DN** entry.
  - **Whole Subtree:** Often referred to as “sub,” selecting this option will search the **Base DN** entry and all its subordinates to any depth.

Refer to [The LDAP Search Operation](#) for more information on search scopes.

## Update

Configure this operation using these parameters:

- **Credential to connect with:** Select or create an [LDAP credential](#) to connect with.
- **DN:** Enter the Distinguished Name (DN) of the entry to update.
- **\*Update Attributes:** Select whether to **Add** new, **Remove** existing, or **Replace** existing attribute.
- Then enter the **Attribute ID/Value** pair you'd like to update.

## Templates and examples

---

## Limit

Use the Limit node to remove items beyond a defined maximum number. You can choose whether n8n takes the items from the beginning or end of the input data.

## Node parameters

Configure this node using the following parameters.

### Max Items

Enter the maximum number of items that n8n should keep. If the input data contains more than this value, n8n removes the items.

### Keep

If the node has to remove items, select where it keeps the input items from:

- **First Items:** Keeps the **Max Items** number of items from the beginning of the input data.
- **Last Items:** Keeps the **Max Items** number of items from the end of the input data.

## Templates and examples

### Related resources

-8<- “\_snippets/integrations/builtin/core-nodes/data-transformation-actions/data-section-link.md”

---

## Local File Trigger node

The Local File Trigger node starts a workflow when it detects changes on the file system. These changes involve a file or folder getting added, changed, or deleted.

### Node parameters

You can choose what event to watch for using the **Trigger On** parameter.

### Changes to a Specific File

The node triggers when the specified file changes.

Enter the path for the file to watch in **File to Watch**.

### Changes Involving a Specific Folder

The node triggers when a change occurs in the selected folder.

Configure these parameters:

- **Folder to Watch:** Enter the path of the folder to watch.
- **Watch for:** Select the type of change to watch for.

## Node options

Use the node **Options** to include or exclude files and folders.

- **Include Linked Files/Folders:** also watch for changes to linked files or folders.
- **Ignore:** files or paths to ignore. n8n tests the whole path, not just the filename. Supports the Anymatch syntax.
- **Max Folder Depth:** how deep into the folder structure to watch for changes.

### Examples for Ignore

Ignore a single file:

```
**/<fileName>.<suffix>
For example, **/myfile.txt
```

Ignore a sub-directory of a directory you're watching:

```
/<directoryName>/
For example, **/myDirectory/**
```

---

## Templates and examples

## Loop Over Items

The Loop Over Items node helps you loop through data when needed.

The node saves the original incoming data, and with each iteration, returns a predefined amount of data through the **loop** output.

When the node execution completes, it combines all of the processed data and returns it through the **done** output.

## When to use the Loop Over Items node

By default, n8n nodes are designed to process a list of input items (with some exceptions, detailed below). Depending on what you're trying to achieve, you often don't need the Loop Over Items node in your workflow. You can learn more about how n8n processes multiple items on the [looping in n8n](#) page.

These links highlight some of the cases where the Loop Over Items node can be useful:

- [Loop until all items are processed](#): describes how the Loop Over Items node differs from normal item processing and when you might want to incorporate this node.
- [Node exceptions](#): outlines specific cases and nodes where you may need to use the Loop Over Items node to manually build looping

logic.

- Avoiding rate limiting: demonstrates how to batch API requests to avoid rate limits from other services.

## Node parameters

### Batch Size

Enter the number of items to return with each call.

### Node options

#### Reset

If turned on, the node will reset with the current input-data newly initialized with each loop. Use this when you want the Loop Over Items node to treat incoming data as a new set of data instead of a continuation of previous items.

For example, you can use the Loop Over Items node with the reset option and an If node to query a paginated service when you don't know how many pages you need in advance. The loop queries pages one at a time, performs any processing, and increments the page number. The loop reset ensures the loop recognizes each iteration as a new set of data. The If node evaluates an exit condition to decide whether to perform another iteration or not.

When enabled, you can adjust the reset conditions by switching the parameter representation from **Fixed** to **Expression**. The results of your expression evaluation determine when the node will reset item processing.

## Templates and examples

### Read RSS feed from two different sources

This workflow allows you to read an RSS feed from two different sources using the Loop Over Items node. You need the Loop Over Items node in the workflow as the RSS Feed Read node only processes the first item it receives. You can also find the workflow on n8n.io.

The example walks through building the workflow, but assumes you are already familiar with n8n. To build your first workflow, including learning how to add nodes to a workflow, refer to Try it out.

The final workflow looks like this:

Copy the workflow file above and paste into your instance, or manually build it by following these steps:

1. Add the manual trigger.
2. Add the Code node.
3. Copy this code into the Code node:

```
js return [{
 json: { url: 'https://medium.com/feed/n8n-io',
 } }, { json: { url:
 'https://dev.to/feed/n8n', } }];
```

4. Add the Loop Over Items node.
5. Configure Loop Over Items: set the batch size to 1 in the **Batch Size** field.
6. Add the RSS Feed Read node.
7. Select **Execute Workflow**. This runs the workflow to load data into the RSS Feed Read node.
8. Configure RSS Feed Read: map `url` from the input to the **URL** field. You can do this by dragging and dropping from the **INPUT** panel, or using this expression: `{{ $json.url }}`.
9. Select **Execute Workflow** to run the workflow and see the resulting data.

### **Check that the node has processed all items**

To check if the node still has items to process, use the following expression: `{{$node["Loop Over Items"].context["noItemsLeft"]}}`. This expression returns a boolean value. If the node still has data to process, the expression returns false, otherwise it returns true.

### **Get the current running index of the node**

To get the current running index of the node, use the following expression: `{{$node["Loop Over Items"].context["currentRunIndex"]}}`.

---

## **Manual Trigger node**

Use this node if you want to start a workflow by selecting **Execute Workflow** and don't want any option for the workflow to run automatically.

Workflows always need a trigger, or start point. Most workflows start with a trigger node firing in response to an external event or the [Schedule Trigger](#) firing on a set schedule.

The Manual Trigger node serves as the workflow trigger for workflows that don't have an automatic trigger.

Use this trigger:

- To test your workflow before you add an automatic trigger of some kind.
- When you don't want the workflow to run automatically.

## **Common issues**

Here are some common errors and issues with the Manual Trigger node and steps to resolve or troubleshoot them.

### **Only one 'Manual Trigger' node is allowed in a workflow**

This error displays if you try to add a Manual Trigger node to a workflow which already includes a Manual Trigger node.

Remove your existing Manual Trigger or edit your workflow to connect that trigger to a different node.

---

## Markdown

The Markdown node converts between Markdown and HTML formats.

## Operations

This node's operations are **Modes**:

- **Markdown to HTML:** Use this mode to convert from Markdown to HTML.
- **HTML to Markdown:** Use this mode to convert from HTML to Markdown.

## Node parameters

- **HTML or Markdown:** Enter the data you want to convert. The field name changes based on which **Mode** you select.
- **Destination Key:** Enter the field you want to put the output in. Specify nested fields using dots, for example `level1.level2.newKey`.

## Node options

The node's **Options** depend on the **Mode** selected.

### Markdown to HTML options

Option	Description	Default
<b>Add Blank To Links</b>	Whether to open links a new window (enabled) or not (disabled).	Disabled
<b>Automatic Linking To URLs</b>	Whether to automatically link to URLs (enabled) or not (disabled). If enabled, n8n converts any string that it identifies as a URL to a link.	Disabled
<b>Backslash Escapes</b>	Whether to allow backslash escaping of HTML tags (enabled) or not (disabled).	Disabled
<b>HTML Tags</b>	When enabled, n8n escapes any < or > prefaced with \. For example, \<div\> renders as &lt;div&gt;.	
<b>Complete HTML Document</b>	Whether to output a complete HTML document (enabled) or an HTML fragment (disabled). A complete HTML document includes the <DOCTYPE HTML> declaration,	Disabled

<b>Customized Header ID</b>	<html> and <body> tags, and the <head> element.	Whether to support custom heading IDs (enabled) or not (disabled). When enabled, you can add custom heading IDs using {header ID here} after the heading text.	Disabled
<b>Emoji Support</b>	Whether to support emojis (enabled) or not (disabled).	Disabled.	
<b>Encode Emails</b>	Whether to transform ASCII character emails into their equivalent decimal entities (enabled) or not (disabled).	Enabled	
<b>Exclude Trailing Punctuation From URLs</b>	Whether to exclude trailing punctuation from automatically linked URLs (enabled) or not (disabled). For use with <b>Automatic Linking To URLs</b> .	Disabled	
<b>GitHub Code Blocks</b>	Whether to enable GitHub Flavored Markdown code blocks (enabled) or not (disabled).	Enabled	
<b>GitHub Compatible Header IDs</b>	Whether to generate GitHub Flavored Markdown heading IDs (enabled) or not (disabled). GitHub Flavored Markdown generates heading IDs with - in place of spaces and removes non-alphanumeric characters.	Disabled	
<b>GitHub Mention Link</b>	Change the link used with <b>GitHub Mentions</b> .	Disabled	
<b>GitHub Mentions</b>	Whether to support tagging GitHub users with @ (enabled) or not (disabled). When enabled, n8n replaces @name with <a href="https://github.com/name">https://github.com/name</a> .	Disabled	
<b>GitHub Task Lists</b>	Whether to support GitHub Flavored Markdown task lists (enabled) or not (disabled).	Disabled	
<b>Header Level Start</b>	Number. Set the start level for headers. For example, changing this field to 2 causes n8n to treat # as <h2>, ## as <h3>, and so on.	1	
<b>Mandatory Space Before Header</b>	Whether to make a space between # and heading text required (enabled) or not (disabled). When enabled, n8n renders a heading written as ##Some header text literally (it doesn't turn it into a heading element)	Disabled	
	Whether n8n should treat		

<b>Middle Word Asterisks</b>	asterisks in words as Markdown (disabled) or render them as literal asterisks (enabled).	Disabled
<b>Middle Word Underscores</b>	Whether n8n should treat underscores in words as Markdown (disabled) or render them as literal underscores (enabled).	Disabled
<b>No Header ID</b>	Disable automatic generation of header IDs (enabled).	Disabled
<b>Parse Image Dimensions</b>	Support setting maximum image dimensions in Markdown syntax (enabled).	Disabled
<b>Prefix Header ID</b>	Define a prefix to add to header IDs.	None
<b>Raw Header ID</b>	Whether to remove spaces, ', and " from header IDs, including prefixes, replacing them with - (enabled) or not (disabled).	Disabled
<b>Raw Prefix Header ID</b>	Whether to prevent n8n from modifying header prefixes (enabled) or not (disabled)	Disabled
<b>Simple Line Breaks</b>	Whether to create line breaks without a double space at the end of a line (enabled) or not (disabled).	Disabled
<b>Smart Indentation Fix</b>	Whether to try to smartly fix indentation problems related to ES6 template strings in indented code blocks (enabled) or not (disabled).	Disabled
<b>Spaces Indented Sublists</b>	Whether to remove the requirement to indent sublists four spaces (enabled) or not (disabled).	Disabled
<b>Split Adjacent Blockquotes</b>	Whether to split adjacent blockquote blocks (enabled) or not (disabled). If you don't enable this, n8n treats quotes (indicated by > at the start of the line) on separate lines as a single blockquote, even when separated by an empty line.	Disabled
<b>Strikethrough</b>	Whether to support strikethrough syntax (enabled) or not (disabled). When enabled, you can add a <b>strikethrough</b> effect using ~~ around the word or phrase.	Disabled
<b>Tables Header ID</b>	Whether to add an ID to table header tags (enabled) or not (disabled).	Disabled
<b>Tables</b>	Whether to support tables	Disabled

<b>Support</b>	(enabled) or not (disabled).
----------------	------------------------------

## HTML to Markdown options

Option	Description	Default
<b>Bullet Marker</b>	Specify the character to use for unordered lists.	*
<b>Code Block Fence</b>	Specify the characters to use for code blocks.	``     **Emphasis Delimiter**   Specify the character.   _     **Global Escape Pattern**   Overrides the default character escape settings. You may want to use Text Replacement Pattern instead.   None     **Ignored Elements**   Ignore given HTML elements, and their children.   None     **Keep Images With Data**   Whether to keep images with data (enabled) or not (disabled). Support files up to 1MB.   Disabled     **Line Start Escape Pattern**   Overrides the default character escape settings. You may want to use Text Replacement Pattern instead.   None     **Max Consecutive New Lines**   Number. Specify the maximum number of consecutive new lines allowed.   3     **Place URLs At The Bottom**   Whether to place URLs at the bottom of the page and format using link reference definitions (enabled) or not

		(disabled).   Disabled     **Strong Delimiter**   Specify the characters for `.
<b>Style For Code Block</b>	Specify the styling for code blocks. Options are <b>Fence</b> and <b>Indented</b> .	Fence
<b>Text Replacement Pattern</b>	Define a text replacement pattern using regex.	None
<b>Treat As Blocks</b>	Specify HTML elements to treat as blocks (surround with blank lines)	None

## Templates and examples

### Parsers

n8n uses the following parsers:

- To convert from HTML to Markdown: [node-html-markdown](#).
- To convert from Markdown to HTML: [Showdown](#). Some options allow you to extend your Markdown with [GitHub Flavored Markdown](#).

---

## MCP Client node

The MCP Client node is a [Model Context Protocol \(MCP\)](#) client that allows you to use the tools that are exposed by an external MCP server.

You can use the MCP Client node to use MCP tools as regular steps in a workflow.

If you want to use MCP tools as tools for an AI Agent, use the [MCP Client Tool node](#) instead.

### Node parameters

Configure the node with the following parameters.

- **Server Transport:** The transport protocol used by the MCP Server endpoint you want to connect to.
- **MCP Endpoint URL:** The URL of the external MCP Server. For example, <https://mcp.notion.com/mcp>.
- **Authentication:** The authentication method for authentication to your MCP server. The MCP Client node supports [bearer](#), generic [header](#), and [OAuth2](#) authentication. Select **None** to attempt to connect without authentication.
- **Tool:** Select the tool to use in the node. The list of tools is

automatically fetched from the external MCP server.

- **Input Mode:**
  - **Manual:** Specify each tool parameter manually.
  - **JSON:** Specify tool parameters as a JSON object. Use this mode for tools with nested parameters.

## Options

- **Convert to Binary:** Whether to convert images and audio to binary data. If false, images and audio are returned as base64 encoded strings.
- **Timeout:** Time in milliseconds to wait for tool calls to finish.

## Templates and examples

### Related resources

To use MCP tools with AI Agents, n8n has the [MCP Client Tool node](#).

n8n also has an [MCP Server Trigger](#) node that allows you to expose n8n tools to external AI Agents.

Refer to the [MCP documentation](#) and [MCP specification](#) for more details about the protocol, servers, and clients.

```
-8<- "_snippets/integrations/builtin/cluster-nodes/tools-link.md"
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

---

## MCP Server Trigger node

Use the MCP Server Trigger node to allow n8n to act as a [Model Context Protocol \(MCP\)](#) server, making n8n tools and workflows available to MCP clients.

### How the MCP Server Trigger node works

The MCP Server Trigger 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.

Unlike conventional [trigger nodes](#), which respond to events and pass their output to the next [connected node](#), the MCP Server Trigger node only connects to and executes [tool nodes](#). Clients can list the available tools and call individual tools to perform work.

You can expose n8n workflows to clients by attaching them with the [Custom n8n Workflow Tool](#) node.

### Node parameters

Use these parameters to configure your node.

## MCP URL

The MCP Server Trigger node has two **MCP URLs**: test and production. n8n displays the URLs at the top of the node panel.

Select **Test URL** or **Production URL** to toggle which URL n8n displays.

- **Test:** n8n registers a test MCP URL when you select **Listen for Test Event** or **Execute workflow**, if the workflow isn't active. When you call the MCP URL, n8n displays the data in the workflow.
- **Production:** n8n registers a production MCP URL when you publish the workflow. When using the production URL, n8n doesn't display the data in the workflow. You can still view workflow data for a production execution: select the **Executions** tab in the workflow, then select the workflow execution you want to view.

## Authentication

You can require authentication for clients connecting to your MCP URL. Choose from these authentication methods:

- Bearer auth
- Header auth

Refer to the [HTTP request credentials](#) for more information on setting up each credential type.

## Path

By default, this field contains a randomly generated MCP URL path, to avoid conflicts with other MCP Server Trigger nodes.

You can manually specify a URL path, including adding route parameters. For example, you may need to do this if you use n8n to prototype an API and want consistent endpoint URLs.

## Templates and examples

### Integrating with Claude Desktop

You can connect to the MCP Server Trigger node from [Claude Desktop](#) by running a gateway to proxy SSE messages to stdio-based servers.

To do so, add the following to your Claude Desktop configuration:

```
{
 "mcpServers": {
 "n8n": {
 "command": "npx",
 "args": [
 "mcp-remote",
 "<MCP_URL>",
 "--header",
 "Authorization: Bearer ${AUTH_TOKEN}"
]
 }
 }
}
```

```
],
 "env": {
 "AUTH_TOKEN": "<MCP_BEARER_TOKEN>"
 }
 }
}
```

Be sure to replace the <MCP\_URL> and <MCP\_BEARER\_TOKEN> placeholders with the values from your MCP Server Trigger node parameters and credentials.

## Limitations

### Configuring the MCP Server Trigger node with webhook replicas

The MCP Server Trigger node relies on Server-Sent Events (SSE) or streamable HTTP, which require the same server instance to handle persistent connections. This can cause problems when running n8n in queue mode depending on your webhook processor configuration:

- If you use queue mode with a **single webhook replica**, the MCP Server Trigger node works as expected.
- If you run **multiple webhook replicas**, you need to route all /mcp\* requests to a single, dedicated webhook replica. Create a separate replica set with one webhook container for MCP requests.  
Afterward, update your ingress or load balancer configuration to direct all /mcp\* traffic to that instance.

## Related resources

n8n also provides an MCP Client Tool node that allows you to connect your n8n AI agents to external tools.

Refer to the MCP documentation and MCP specification for more details about the protocol, servers, and clients.

## Common issues

Here are some common errors and issues with the MCP Server Trigger node and steps to resolve or troubleshoot them.

### Running the MCP Server Trigger node with a reverse proxy

When running n8n behind a reverse proxy like nginx, you may experience problems if the MCP endpoint isn't configured for SSE or streamable HTTP.

Specifically, you need to disable proxy buffering for the endpoint. Other items you might want to adjust include disabling gzip compression (n8n handles this itself), disabling chunked transfer encoding, and setting the Connection to an empty string to remove it

from the forwarded headers. Explicitly disabling these in the MCP endpoint ensures they're not inherited from other places in your nginx configuration.

An example nginx location block for serving MCP traffic with these settings may look like this:

```
location /mcp/ {
 proxy_http_version 1.1;
 proxy_buffering off;
 gzip off;
 chunked_transfer_encoding off;

 proxy_set_header Connection '';
 # The rest of your proxy headers and settings
 # . . .
}
```

---

## Merge

Use the Merge node to combine data from multiple streams, once data of all streams is available.

### Node parameters

You can specify how the Merge node should combine data from different data streams by choosing a **Mode**:

#### Append

Keep data from all inputs. Choose a **Number of Inputs** to output items of each input, one after another. The node waits for the execution of all connected inputs.

[Image: Sample Append mode inputs and output. Two separate data sources are on the left, one with items A, B, C and one with items D, E, F. The final data source combines both and lists A, B, C, D, E, F.]  
Append mode inputs and output

#### Combine

Combine data from two inputs. Select an option in **Combine By** to determine how you want to merge the input data.

#### Matching Fields

Compare items by field values. Enter the fields you want to compare in **Fields to Match**.

n8n's default behavior is to keep matching items. You can change this using the **Output Type** setting:

- **Keep Matches:** Merge items that match. This is like an inner join.

- **Keep Non-Matches:** Merge items that don't match.
- **Keep Everything:** Merge items together that do match and include items that don't match. This is like an outer join.
- **Enrich Input 1:** Keep all data from Input 1, and add matching data from Input 2. This is like a left join.
- **Enrich Input 2:** Keep all data from Input 2, and add matching data from Input 1. This is like a right join.

[Image: Sample Combine mode inputs and output. Two separate data sources are on the left. The final data source combines these data sources by matching fields.]

Combine by Matching Fields mode inputs and output

## Position

Combine items based on their order. The item at index 0 in Input 1 merges with the item at index 0 in Input 2, and so on.

[Image: Sample Combine mode inputs and output. Two separate data sources are on the left. The final data source combines these data sources by index position.]

Combine by Position mode inputs and output

## All Possible Combinations

Output all possible item combinations, while merging fields with the same name.

[Image: Sample Combine mode inputs and output. Two separate data sources are on the left. The final data source combines these data sources by all possible combinations.]

Combine by All Possible Combinations mode inputs and output

## Combine mode options

When merging data by **Mode > Combine**, you can set these **Options**:

- **Clash Handling:** Choose how to merge when data streams clash, or when there are sub-fields. Refer to [Clash handling](#) for details.
- **Fuzzy Compare:** Whether to tolerate type differences when comparing fields (enabled), or not (disabled, default). For example, when you enable this, n8n treats "3" and 3 as the same.
- **Disable Dot Notation:** This prevents accessing child fields using parent.child in the field name.
- **Multiple Matches:** Choose how n8n handles multiple matches when comparing data streams.
  - **Include All Matches:** Output multiple items if there are multiple matches, one for each match.
  - **Include First Match Only:** Keep the first item per match and discard the remaining multiple matches.
- **Include Any Unpaired Items:** Choose whether to keep or discard unpaired items when merging by position. The default behavior is to leave out the items without a match.

## *Clash Handling*

-8<- “\_snippets/integrations/builtin/core-nodes/merge/field-value-clash.md”

## SQL Query

Write a custom SQL Query to merge the data.

Example:

```
SELECT * FROM input1 LEFT JOIN input2 ON input1.name = input2.id
```

Data from previous nodes are available as tables and you can use them in the SQL query as input1, input2, input3, and so on, based on their order. Refer to [AlaSQL GitHub page](#) for a full list of supported SQL statements.

## Choose Branch

Choose which input to keep. This option always waits until the data from both inputs is available. You can choose to **Output**:

- The **Input 1 Data**
- The **Input 2 Data**
- **A Single, Empty Item**

The node outputs the data from the chosen input, without changing it.

## Templates and examples

### Merging data streams with uneven numbers of items

The items passed into Input 1 of the Merge node will take precedence. For example, if the Merge node receives five items in Input 1 and 10 items in Input 2, it only processes five items. The remaining five items from Input 2 aren't processed.

### Branch execution with If and Merge nodes

```
-8<- "_snippets/integrations/builtin/core-nodes/merge/if-merge-branch-execution.md"
```

### Try it out: A step by step example

Create a workflow with some example input data to try out the Merge node.

#### Set up sample data using the Code nodes

1. Add a Code node to the canvas and connect it to the Start node.
2. Paste the following JavaScript code snippet in the **JavaScript Code** field:

```
return [
{
 json: {
 name: 'Stefan',
```

```

 language: 'de',
 },
},
{
 json: {
 name: 'Jim',
 language: 'en',
 }
},
{
 json: {
 name: 'Hans',
 language: 'de',
 }
}
];

```

3. Add a second Code node, and connect it to the Start node.
4. Paste the following JavaScript code snippet in the **JavaScript Code** field:

```

return [
 {
 json: {
 greeting: 'Hello',
 language: 'en',
 }
 },
 {
 json: {
 greeting: 'Hallo',
 language: 'de',
 }
 }
];

```

## Try out different merge modes

Add the Merge node. Connect the first Code node to **Input 1**, and the second Code node to **Input 2**. Run the workflow to load data into the Merge node.

The final workflow should look like this:

Now try different options in **Mode** to see how it affects the output data.

### Append

Select **Mode > Append**, then select **Execute step**.

Your output in table view should look like this:

---

name	language	greeting
Stefan	de	
Jim	en	
Hans	de	
	en	Hello

de                    Hallo

---

### Combine by Matching Fields

You can merge these two data inputs so that each person gets the correct greeting for their language.

1. Select **Mode > Combine**.
2. Select **Combine by > Matching Fields**.
3. In both **Input 1 Field** and **Input 2 Field**, enter language. This tells n8n to combine the data by matching the values in the language field in each data set.
4. Select **Execute step**.

Your output in table view should look like this:

name	language	greeting
Stefan	de	Hallo
Jim	en	Hello
Hans	de	Hallo

### Combine by Position

Select **Mode > Combine, Combine by > Position**, then select **Execute step**.

Your output in table view should look like this:

name	language	greeting
Stefan	en	Hello
Jim	de	Hallo

### *Keep unpaired items*

If you want to keep all items, select **Add Option > Include Any Unpaired Items**, then turn on **Include Any Unpaired Items**.

Your output in table view should look like this:

name	language	greeting
Stefan	en	Hello
Jim	de	Hallo
Hans	de	

### Combine by All Possible Combinations

Select **Mode > Combine, Combine by > All Possible Combinations**, then select **Execute step**.

Your output in table view should look like this:

name	language	greeting
Stefan	en	Hello
Stefan	de	Hallo

Jim	en	Hello
Jim	de	Hallo
Hans	en	Hello
Hans	de	Hallo

---

## n8n

A node to integrate with n8n itself. This node allows you to consume the [n8n API](#) in your workflows.

Refer to the [n8n REST API documentation](#) for more information on using the n8n API. Refer to [API endpoint reference](#) for working with the API endpoints directly.

## Operations

- Audit
  - [Generate a security audit](#)
- Credential
  - [Create a credential](#)
  - [Delete a credential](#)
  - [Get Schema](#): Use this operation to get credential data schema for type
- Execution
  - [Get an execution](#)
  - [Get Many executions](#)
  - [Delete an execution](#)
- Workflow
  - [Publish a workflow](#)
  - [Create a workflow](#)
  - [Deactivate a workflow](#)
  - [Delete a workflow](#)
  - [Get a workflow](#)
  - [Get Many workflows](#)
  - [Update a workflow](#)

### Generate audit

This operation has no parameters. Configure it with these options:

- **Categories**: Select the risk categories you want the audit to include. Options include:
  - **Credentials**
  - **Database**
  - **Filesystem**
  - **Instance**
  - **Nodes**
- **Days Abandoned Workflow**: Use this option to set the number of days without execution after which a workflow should be considered abandoned. Enter a number of days. The default is 90.

## Create credential

Configure this operation with these parameters:

- **Name:** Enter the name of the credential you'd like to create.
- **Credential Type:** Enter the credential's type. The available types depend on nodes installed on the n8n instance. Some built-in types include `githubApi`, `notionApi`, and `slackApi`.
- **Data:** Enter a valid JSON object with the required properties for this **Credential Type**. To see the expected format, use the **Get Schema** operation.

## Delete credential

Configure this operation with this parameter:

- **Credential ID:** Enter the ID of the credential you want to delete.

## Get credential schema

Configure this operation with this parameter:

- **Credential Type:** Enter the credential's type. The available types depend on nodes installed on the n8n instance. Some built-in types include `githubApi`, `notionApi`, and `slackApi`.

## Get execution

Configure this operation with this parameter:

- **Execution ID:** Enter the ID of the execution you want to retrieve.

## Get execution option

You can further configure this operation with this **Option**:

- **Include Execution Details:** Use this control to set whether to include the detailed execution data (turned on) or not (turned off).

## Get many executions

Configure this operation with these parameters:

- **Return All:** Set whether to return all results (turned on) or whether to limit the results to the entered **Limit** (turned on).
- **Limit:** Set the number of results to return if the **Return All** control is turned off.

## Get many executions filters

You can further configure this operation with these **Filters**:

- **Workflow:** Filter the executions by workflow. Options include:
  - **From list:** Select a workflow to use as a filter.

- **By URL:** Enter a workflow URL to use as a filter.
- **By ID:** Enter a workflow ID to use as a filter.
- **Status:** Filter the executions by status. Options include:
  - **Error**
  - **Success**
  - **Waiting**

## Get many execution options

You can further configure this operation with this **Option**:

- **Include Execution Details:** Use this control to set whether to include the detailed execution data (turned on) or not (turned off).

## Delete execution

Configure this operation with this parameter:

- **Execution ID:** Enter the ID of the execution you want to delete.

## Activate, deactivate, delete, and get workflow

The **Activate**, **Deactivate**, **Delete**, and **Get** workflow operations all include the same parameter for you to select the **Workflow** you want to perform the operation on. Options include:

- **From list:** Select the workflow from the list.
- **By URL:** Enter the URL of the workflow.
- **By ID:** Enter the ID of the workflow.

## Create workflow

Configure this operation with this parameter:

- **Workflow Object:** Enter a valid JSON object with the new workflow's details. The object requires these fields:
  - name
  - nodes
  - connections
  - settings

Refer to [n8n API reference](#) for more information.

## Get many workflows

Configure this operation with these parameters:

- **Return All:** Set whether to return all results (turned on) or whether to limit the results to the entered **Limit** (turned on).
- **Limit:** Set the number of results to return if the **Return All** control is turned off.

## Get many workflows filters

You can further configure this operation with these **Filters**:

- **Return Only Active Workflows:** Select whether to return only active workflows (turned on) or active and inactive workflows (turned off).
- **Tags:** Enter a comma-separated list of tags the returned workflows must have.

## Update workflow

Configure this operation with these parameters:

- **Workflow:** Select the workflow you want to update. Options include:
  - **From list:** Select the workflow from the list.
  - **By URL:** Enter the URL of the workflow.
  - **By ID:** Enter the ID of the workflow.
- **Workflow Object:** Enter a valid JSON object to update the workflow with. The object requires these fields:
  - name
  - nodes
  - connections
  - settings

Refer to the [n8n API | Update a workflow documentation](#) for more information.

## Templates and examples

---

## n8n Form node

Use the n8n Form node to create user-facing forms with multiple steps. You can add other nodes with custom logic between to process user input. You must start the workflow with the [n8n Form Trigger node](#).

## Setting up the node

### Set default selections with query parameters

You can set the initial values for fields by using [query parameters](#) with the initial URL provided by the [n8n Form Trigger node](#). Every page in the form receives the same query parameters sent to the n8n Form Trigger node URL.

When using query parameters, [percent-encode](#) any field names or values that use special characters. This ensures n8n uses the initial values for the given fields. You can use tools like [URL Encode/Decode](#) to format your query parameters using percent-encoding.

As an example, imagine you have a form with the following properties:

- Production URL: `https://my-account.n8n.cloud/form/my-form`
- Fields:
  - name: Jane Doe
  - email: jane.doe@example.com

With query parameters and percent-encoding, you could use the following URL to set initial field values to the data above:

```
https://my-account.n8n.cloud/form/my-form?
email=jane.doe%40example.com&name=Jane%20Doe
```

Here, percent-encoding replaces the at-symbol (@) with the string %40 and the space character () with the string %20. This will set the initial value for these fields no matter which page of the form they appear on.

## Displaying custom HTML

You can display custom HTML on your form by adding a **Custom HTML** field to your form. This provides an **HTML** box where you can insert arbitrary HTML code to display as part of the form page.

You can use the HTML field to enrich your form page by including things like links, images, videos, and more. n8n will render the content with the rest of the form fields in the normal document flow.

Because custom HTML content is read-only, these fields aren't included in the form output data by default. To include the raw HTML content in the node output, provide a name for the data using the **Element Name** field.

The HTML field doesn't support `<script>`, `<style>`, or `<input>` elements.

If you're using the Form Ending Page Type, you can fully customize the final page that you send users (including the use of `<script>`, `<style>`, and `<input>` elements) by selecting the **On n8n Form Submission** parameter to **Show Text**.

## Including hidden fields

It's possible to include fields in a form without displaying them to users. This is useful when you want to pass extra data to the form that doesn't require interactive user input.

To add fields that won't show up on the form, use the **Hidden Field** form element. There, you can define the **Field Name** and optionally provide a default value by filling out the **Field Value**.

When serving the form, you can pass values for hidden fields using query parameters.

## Defining the form using JSON

Use **Define Form > Using JSON** to define the fields of your form with a JSON array of objects. Each object defines a single field by using a combination of these keys:

- **fieldLabel**: The label that appears above the input field on the rendered form.

- **fieldName**: The field name that is used in the output of the Form node and to reference the field in expressions.
- **fieldType**: Choose from checkbox, date, dropdown, email, file, hiddenField, html, number, password, radio, text, or textarea.
  - Use date to include a date picker in the form. Refer to [Date and time with Luxon](#) for more information on formatting dates.
  - When using dropdown, set the choices with **fieldOptions** (reference the example below). By default, the dropdown is single-choice. To make it multiple-choice, set **multiselect** to true. As an alternative, you can use checkbox or radio together with **fieldOptions** too.
  - When using file, set **multipleFiles** to true to allow users to select more than one file. To define the file types to allow, set **acceptFileTypes** to a string containing a comma-separated list of file extensions (reference the example below).
  - Use **hiddenField** to add a hidden field to your form. Refer to [Including hidden fields](#) for more information.
  - Use **html** to display custom HTML on your form. Refer to [Displaying custom HTML](#) for more information.
- **placeholder**: Specify placeholder data for the field. You can use this for every **fieldType** except dropdown, checkbox, radio, date, and file.
- **defaultValue**: Specify a value that will be pre-filled or pre-selected in the form element. You can use this for every **fieldType** except password, html, hiddenField and file.
- **requiredField**: Require users to complete this field on the form.

An example JSON that shows the general format required and the keys available:

```
// Use the "requiredField" key on any field to mark it as mandatory
// Use the "placeholder" key to specify placeholder data for all
fields
// except 'dropdown', 'checkbox', 'radio', 'date' and 'file'
// Use the "defaultValue" key to pre-fill a form field or pre-select
a
// value in 'checkbox', 'radio' or 'dropdown' fields

[
 {
 "fieldLabel": "Date Field",
 "fieldType": "date",
 "formatDate": "mm/dd/yyyy", // how to format received date in
n8n
 "requiredField": true
 },
 {
 "fieldLabel": "Dropdown Options",
 "fieldType": "dropdown",
 "fieldOptions": {
 "values": [
 {
 "option": "option 1"
 },
 {
 "option": "option 2"
 }
],
 "defaultValue": "option 1",
 "requiredField": true
 },
 }
]
```

```
{
 "fieldLabel": "Multiselect",
 "fieldType": "dropdown",
 "fieldOptions": {
 "values": [
 {
 "option": "option 1"
 },
 {
 "option": "option 2"
 }
]
 },
 "multiselect": true // setting to true allows multi-select
},
{
 "fieldLabel": "Email",
 "fieldType": "email",
 "placeholder": "me@mail.com"
},
{
 "fieldLabel": "File",
 "fieldType": "file",
 "multipleFiles": true, // setting to true allows multiple files
 selection
 "acceptFileTypes": ".jpg, .png" // allowed file types
},
{
 "fieldLabel": "Number",
 "fieldType": "number"
},
{
 "fieldLabel": "Password",
 "fieldType": "password"
},
{
 // "fieldType": "text" can be omitted since it's the default
 type
 "fieldLabel": "Text"
},
{
 "fieldLabel": "Textarea",
 "fieldType": "textarea",
 "defaultValue": "Lorem ipsum."
},
{
 "fieldType": "html",
 "elementName": "content", // Optional field. It can be used to
 include the html in the output.
 "html": "<div>Custom element</div>"
},
{
 "fieldLabel": "Checkboxes",
 "fieldType": "checkbox",
 "fieldOptions": {
 "values": [
 {
 "option": "option 1"
 },
 {
 "option": "option 2"
 }
]
 }
}
```

```

 "option": "option 2"
 }
]
},
"defaultValue": ["option 1", "option 2"]
},
{
"fieldLabel": "Radio",
"fieldType": "radio",
"fieldOptions": {
"values": [
{
"option": "option 1"
},
{
"option": "option 2"
}
]
}
},
{
"fieldLabel": "hidden label",
"fieldType": "hiddenField",
"fieldValue": "extra form data"
}
]

```

## Form Ending

Use the **Form Ending** Page Type to end a form and either show a completion page, redirect the user to a URL, or display custom HTML or text. Only one Form Ending page displays per execution, even when n8n executes multiple branches that contain Form Ending nodes.

Choose between these options when using **On n8n Form Submission**:

- **Show Completion Screen:** Shows users a final screen to confirm that they submitted the form.
  - Fill in **Completion Title** to set the h1 title on the form.
  - n8n displays the **Completion Message** as a subtitle below the main h1 title on the form. Use <br> to add a line break.
  - Select **Add option** and fill in **Completion Page Title** to set the page's title in the browser tab.
- **Redirect to URL:** Redirect the user to a specified URL when the form completes.
  - Fill in the **URL** field with the page you want to redirect to when users complete the form.
- **Show Text:** Display a final page defined by arbitrary plain text and HTML.
  - Fill in the **Text** field with the HTML or plain text content you wish to show.
- **Return Binary File:** Return a binary file upon completion.
  - Fill in **Completion Title** to set the h1 title on the form.
  - n8n displays the **Completion Message** as a subtitle below the main h1 title on the form. Use <br> to add a line break.
  - Provide the **Input Data Field Name** containing the binary file to return to the user.

## Forms with branches

The n8n Form node executes and displays its associated form page whenever it receives data from a previous node. When building forms with n8n, to avoid confusion, it's important to understand how forms behave when branching occurs.

### Workflows with mutually exclusive branches

Form workflows containing mutually exclusive branches work as expected. n8n will execute a single branch according to the submitted data and conditions you outline. As it executes, n8n will display each page in the branch, ending with an n8n Form node with the **Form Ending** page type.

This workflow demonstrates mutually exclusive branching. Each selection can only execute a single branch.

### Workflows that may execute multiple branches

Form workflows that send data to multiple branches at the same time require more care. When multiple branches receive data during an execution (for example, from a switch node), n8n executes each branch that receives data sequentially. Upon reaching the end of one branch, the execution will move to the next branch with data.

n8n only executes a single **Form Ending** n8n Form node for each execution. When multiple branches of a form workflow receive data, n8n ignores all Form Ending nodes except for the one associated with the final branch.

This workflow may execute more than one branch during an execution. Here, n8n executes all valid branches sequentially. This impacts which n8n Form nodes n8n executes (in particular, which **Form Ending** node displays):

## Node options

Select **Add Option** to view more configuration options:

- **Form Title:** The title for your form. n8n displays the **Form Title** as the webpage title and main h1 title on the form.
- **Form Description:** The description for your form. n8n displays the **Form Description** as a subtitle below the main h1 title on the form. This field supports HTML. Use <br> to add a line break. The Form Description also populates the HTML meta description for the page.
- **Button Label:** The label to use for your form's submit button. n8n displays the **Button Label** as the name of the submit button.
- **Custom Form Styling:** Override the default styling of the public form interface with CSS. The field pre-populates with the default styling so you can change only what you need to.
- **Completion Page Title:** The title for the final completion page of the form.

## Running the node

## Build and test workflows

While building or testing a workflow, use the **Test URL** in the [n8n Form Trigger node](#). Using a test URL ensures that you can view the incoming data in the editor UI, which is useful for debugging.

There are two ways to test:

- Select **Execute Step**. n8n opens the form. When you submit the form, n8n runs the node and any previous nodes, but not the rest of the workflow.
- Select **Execute Workflow**. n8n opens the form. When you submit the form, n8n runs the workflow.

## Production workflows

When your workflow is ready, switch to using the n8n Form Trigger's **Production URL** by opening the trigger node and selecting the **Production URL** in the **From URLs** selector. You can then publish your workflow, and n8n runs it automatically when a user submits the form.

When working with a production URL, ensure that you have saved and published the workflow. Data flowing through the Form trigger isn't visible in the editor UI with the production URL.

## Templates and examples

---

## n8n Form Trigger node

Use the n8n Form trigger to start a workflow when a user submits a form, taking the input data from the form. The node generates the form web page for you to use.

You can add more pages to continue the form with the [n8n Form](#) node.

## Build and test workflows

While building or testing a workflow, use the **Test URL**. Using a test URL ensures that you can view the incoming data in the editor UI, which is useful for debugging.

There are two ways to test:

- Select **Execute Step**. n8n opens the form. When you submit the form, n8n runs the node, but not the rest of the workflow.
- Select **Execute Workflow**. n8n opens the form. When you submit the form, n8n runs the workflow.

## Production workflows

When your workflow is ready, switch to using the **Production URL**. You can then publish your workflow, and n8n runs it automatically when a user submits the form.

When working with a production URL, ensure that you have saved and published the workflow. Data flowing through the Form trigger isn't visible in the editor UI with the production URL.

## Set default selections with query parameters

You can set the initial values for fields by using query parameters with the initial URL provided by the n8n Form Trigger. Every page in the form receives the same query parameters sent to the n8n Form Trigger URL.

When using query parameters, percent-encode any field names or values that use special characters. This ensures n8n uses the initial values for the given fields. You can use tools like URL Encode/Decode to format your query parameters using percent-encoding.

As an example, imagine you have a form with the following properties:

- Production URL: <https://my-account.n8n.cloud/form/my-form>
- Fields:
  - name: Jane Doe
  - email: jane.doe@example.com

With query parameters and percent-encoding, you could use the following URL to set initial field values to the data above:

```
https://my-account.n8n.cloud/form/my-form?
email=jane.doe%40example.com&name=Jane%20Doe
```

Here, percent-encoding replaces the at-symbol (@) with the string %40 and the space character () with the string %20. This will set the initial value for these fields no matter which page of the form they appear on.

## Node parameters

These are the main node configuration fields:

### Authentication

- **Basic Auth**
- **None**

### Using basic auth

To configure this credential, you'll need:

- The **Username** you use to access the app or service your HTTP Request is targeting.
- The **Password** that goes with that username.

### Form URLs

The Form Trigger node has two URLs: **Test URL** and **Production URL**. n8n displays the URLs at the top of the node panel. Select **Test URL** or **Production URL** to toggle which URL n8n displays.

[Image: Screenshot of the form URLs]

- **Test URL:** n8n registers a test webhook when you select **Execute Step** or **Execute Workflow**, if the workflow isn't active. When you call the URL, n8n displays the data in the workflow.
- **Production URL:** n8n registers a production webhook when you publish the workflow. When using the production URL, n8n doesn't display the data in the workflow. You can still view workflow data for a production execution. Select the **Executions** tab in the workflow, then select the workflow execution you want to view.

## Form Path

Set a custom slug for the form.

## Form Title

Enter the title for your form. n8n displays the **Form Title** as the webpage title and main h1 title on the form.

## Form Description

Enter the description for your form. n8n displays the **Form Description** as a subtitle below the main h1 title on the form. Use \n or <br> to add a line break.

## Form Elements

Create the question fields for your form. Select **Add Form Element** to add a new field.

Every field has the following settings:

- **Field Label:** Enter the label that appears above the input field on the rendered form.
- **Field Name:** This name is used in the output of the Form Trigger node. Use it to reference a form field in downstream nodes.
- **Element Type:** Choose from **Checkboxes**, **Custom HTML**, **Date**, **Dropdown**, **Email**, **File**, **Hidden Field**, **Number**, **Password**, **Radio Buttons**, **Text**, or **Textarea**.
  - Select **Checkboxes** to include checkbox elements in the form. By default, there is no limit on how many checkboxes a form user can select. You can set the limit by specifying a value for the **Limit Selection** option as **Exact Number**, **Range**, or **Unlimited**.
  - Select **Custom HTML** to insert arbitrary HTML.
    - You can include elements like links, images, video, and more. You can't include <script>, <style>, or <input> elements.
    - By default, Custom HTML fields aren't included in the node output. To include the Custom HTML content in the output, fill out the associated **Element Name** field.
  - Select **Date** to include a date picker in the form. Refer to [Date and time with Luxon](#) for more information on formatting dates.

- Select **Dropdown List** > **Add Field Option** to add multiple options. By default, the dropdown is single-choice. To make it multiple-choice, turn on **Multiple Choice**.
- Select **Radio Buttons** to include radio button elements in the form.
- Select **Hidden Field** to include a form element without displaying it on the form. You can set a default value using the **Field Value** parameter or pass values for the field using query parameters.
- **Placeholder:** Define a sample text to display inside compatible form elements. Placeholders are supported in **Email**, **Number**, **Password**, **Text** and **Textarea**.
- **Default value:** Define a default value that will be pre-filled or pre-selected in compatible form elements. Default values are supported in all form elements except **Custom HTML**, **File**, **Hidden Field**, and **Password**.
- **Required Field:** Turn on to require users to complete this field on the form.

## Respond When

Choose when n8n sends a response to the form submission. You can respond when:

- **Form Is Submitted:** Send a response to the user as soon as they submit the form.
- **Workflow Finishes:** Use this if you want the workflow to complete its execution before you send a response to the user. If the workflow errors, it sends a response to the user telling them there was a problem submitting the form.

## Node options

Select **Add Option** to view more configuration options:

- **Append n8n Attribution:** Turn off to hide the **Form automated with n8n** attribute at the bottom of the form.
- **Button Label:** The label to use for your form's submit button. n8n displays the **Button Label** as the name of the submit button.
- **Form Path:** The final segment of the form's URL, for both testing and production. Replaces the automatically generated UUID as the final component.
- **Ignore Bots:** Turn on to ignore requests from bots like link previewers and web crawlers.
- **Use Workflow Timezone:** Turn on to use the timezone in the Workflow settings instead of UTC (default). This affects the value of the submittedAt timestamp in the node output.
- **Custom Form Styling:** Override the default styling of the public form interface with CSS. The field pre-populates with the default styling so you can change only what you need to.

## Customizing Form Trigger node behavior

### Format response text with line breaks

You can use one of the following methods to add line breaks to form response text:

- Use HTML formatting instead of plain text in the formSubmittedText field
- Replace newline characters (\n) with HTML break tags (<br>)
- before sending the response
- Consider using a custom HTML response page if you need more formatting control

## Restrict form access with authentication

You can use one of the following options to add authentication to your form:

- Use the OTP (One-Time Password) field with TOTP node validation for token-based authentication
- Add a Wait node with form authentication as a secondary form page
- Store hashed passwords in a database and compare against form submissions for validation
- Use external authentication providers like Google Forms if you need advanced authentication

## Templates and examples

---

## n8n Trigger node

The n8n Trigger node triggers when the current workflow updates or activates, or when the n8n instance starts or restarts. You can use the n8n Trigger node to notify when these events occur.

### Node parameters

The node includes a single parameter to identify the **Events** that should trigger it. Choose from these events:

- **Active Workflow Updated:** If you select this event, the node triggers when this workflow is updated.
- **Instance started:** If you select this event, the node triggers when the n8n instance starts or restarts.
- **Workflow Activated:** If you select this event, the node triggers when this workflow is activated.

You can select one or more of these events.

## Templates and examples

---

## No Operation, do nothing

Use the No Operation, do nothing node when you don't want to perform any operations. The purpose of this node is to make the workflow easier to read and understand where the flow of data stops.

This can help others visually get a better understanding of the workflow.

## Templates and examples

---

# Read/Write Files from Disk

Use the Read/Write Files from Disk node to read and write files from/to the machine where n8n is running.

## Operations

- **Read File(s) From Disk:** Use this operation to retrieve one or more files from the computer that runs n8n.
- **Write File to Disk:** Use this operation to create a binary file on the computer that runs n8n.

Refer to the sections below for more information on configuring the node for each operation.

## Read File(s) From Disk

Configure this operation with these parameters:

- **File(s) Selector:** Enter the path of the file you want to read.
  - To enter multiple files, enter a page path pattern. You can use these characters to define a path pattern:
    - \*: Matches any character zero or more times, excluding path separators.
    - \*\*: Matches any character zero or more times, include path separators.
    - ?: Matches any character except for path separators one time.
    - []: Matches any characters inside the brackets. For example, [abc] would match the characters a, b, or c, and nothing else.

Refer to [Picomatch's Basic globbing](#) documentation for more information on these characters and their expected behavior.

## Read File(s) From Disk options

You can also configure this operation with these **Options**:

- **File Extension:** Enter the extension for the file in the node output.
- **File Name:** Enter the name for the file in the node output.
- **MIME Type:** Enter the file's MIME type in the node output. Refer to [Common MIME types](#) for a list of file extensions and their MIME types.
- **Put Output File in Field:** Enter the name of the field in the output data to contain the file.

## Write File to Disk

Configure this operation with these parameters:

- **File Path and Name:** Enter the destination for the file, the file's name, and the file's extension.
- **Input Binary Field:** Enter the name of the field in the node input data that will contain the binary file.

### Write File to Disk options

You can also configure this operation with these **Options**:

This operation includes a single option, whether to **Append** data to an existing file instead of creating a new one (turned on) or to create a new file instead of appending to existing (turned off).

## Templates and examples

### File locations

If you run n8n in Docker, your command runs in the n8n container and not the Docker host.

This node looks for files relative to the n8n install path. n8n recommends using absolute file paths to prevent any errors.

---

## Remove Duplicates node

Use the Remove Duplicates node to identify and delete items that are:

- identical across all fields or a subset of fields in a single execution
- identical to or surpassed by items seen in previous executions

This is helpful in situations where you can end up with duplicate data, such as a user creating multiple accounts, or a customer submitting the same order multiple times. When working with large datasets it becomes more difficult to spot and remove these items.

By comparing against data from previous executions, the Remove Duplicates node can delete items seen in earlier executions. It can also ensure that new items have a later date or a higher value than previous values.

## Operation modes

The remove duplication node works differently depending on the value of the **operation** parameter:

- **Remove Items Repeated Within Current Input:** Identify and remove duplicate items in the current input across all fields or a subset of fields.
- **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.

## Remove Items Repeated Within Current Input

When you set the “Operations” field to **Remove Items Repeated Within Current Input**, the Remove Duplicate node identifies and removes duplicate items in the current input. It can do this across all fields, or within a subset of fields.

### Remove Items Repeated Within Current Input parameters

When using the **Remove Items Repeated Within Current Input** operation, the following parameter is available:

- **Compare**: Select which fields of the input data n8n should compare to check if they’re the same. The following options are available:
  - **All Fields**: Compares all fields of the input data.
  - **All Fields Except**: Enter which input data fields n8n should exclude from the comparison. You can provide multiple values separated by commas.
  - **Selected Fields**: Enter which input data fields n8n should include in the comparison. You can provide multiple values separated by commas.

### Remove Items Repeated Within Current Input options

If you choose **All Fields Except** or **Selected Fields** as your compare type, you can add these options:

- **Disable Dot Notation**: Set whether to use dot notation to reference child fields in the format parent.child (turned off) or not (turn on).
- **Remove Other Fields**: Set whether to remove any fields that aren’t used in the comparison (turned on) or not (turned off).

## Remove Items Processed in Previous Executions

When you set the “Operation” field to **Remove Items Processed in Previous Executions**, the Remove Duplicate node compares items in the current input to items from previous executions.

### Remove Items Processed in Previous Executions parameters

When using the **Remove Items Processed in Previous Executions** operation, the following parameters are available:

- **Keep Items Where**: Select how n8n decides which items to keep. The following options are available:
  - **Value Is New**: n8n removes items if their value matches items from earlier executions.
  - **Value Is Higher than Any Previous Value**: n8n removes items if the current value isn’t higher than previous values.
  - **Value Is a Date Later than Any Previous Date**: n8n removes date items if the current date isn’t later than previous dates.

- **Value to Dedupe On:** The input field or fields to compare. The option you select for the **Keep Items Where** parameter determines the exact format you need:
  - When using **Value Is New**, this must be an input field or combination of fields with a unique ID.
  - When using **Value Is Higher than Any Previous Value**, this must be an input field or combination of fields that has an incremental value.
  - When using **Value Is a Date Later than Any Previous Date**, this must be an input field that has a date value in ISO format.

### **Remove Items Processed in Previous Executions options**

When using the **Remove Items Processed in Previous Executions** operation, the following option is available:

- **Scope:** Sets how n8n stores and uses the deduplication data for comparisons. The following options are available:
  - **Node:** (default) Stores the data for this node independently from other Remove Duplicates instances in the workflow. When you use this scope, you can [clear the duplication history](#) for this node instance without affecting other nodes.
  - **Workflow:** Stores the duplication data at the workflow level. This shares duplication data with any other Remove Duplicate nodes set to use “workflow” scope. n8n will still manage the duplication data for other Remove Duplicate nodes set to “node” scope independently.

When you select **Value Is New** as your **Keep Items Where** choice, this option is also available:

- **History Size:** The number of items for n8n to store to track duplicates across executions. The value of the **Scope** option determines whether this history size is specific to this individual Remove Duplicate node instance or shared with other instances in the workflow. By default, n8n stores 10,000 items.

### **Clear Deduplication History**

When you set the “Operation” field to **Clear Deduplication History**, the Remove Duplicates node manages and clears the stored items from previous executions. This operation doesn’t affect any items in the current input. Instead, it manages the database of items that the “Remove Items Processed in Previous Executions” operation uses.

### **Clear Deduplication History parameters**

When using the **Clear Deduplication History** operation, the following parameter is available:

- **Mode:** How you want to manage the key / value items stored in the database. The following option is available:
  - **Clean Database:** Deletes all duplication data stored in the database. This resets the duplication database to its original state.

### **Clear Deduplication History options**

When using the **Clear Deduplication History** operation, the following option is available:

- **Scope**: Sets the scope n8n uses when managing the duplication database.
  - **Node**: (default) Manages the duplication database specific to this Remove Duplicates node instance.
  - **Workflow**: Manages the duplication database shared by all Remove Duplicate node instances that use workflow scope.

## Templates and examples

For templates using the Remove Duplicates node and examples of how to use it, refer to [Templates and examples](#).

## Related resources

-8<- “\_snippets/integrations/builtin/core-nodes/data-transformation-actions/data-section-link.md”

---

## Templates and examples

Here are some templates and examples for the [Remove Duplicates node](#).

## Templates

## Set up sample data using the Code node

Create a workflow with some example input data to try out the Remove Duplicates node.

1. Add a Code node to the canvas and connect it to the Manual Trigger node.
2. In the Code node, set **Mode** to **Run Once for Each Item** and **Language** to **JavaScript**.
3. Paste the following JavaScript code snippet in the **JavaScript** field:

```
let data =[];

return {
 data: [
 { id: 1, name: 'Taylor Swift', job: 'Pop star', last_updated: '2024-09-20T10:12:43.493Z' },
 { id: 2, name: 'Ed Sheeran', job: 'Singer-songwriter', last_updated: '2024-10-05T08:30:59.493Z' },
 { id: 3, name: 'Adele', job: 'Singer-songwriter', last_updated: '2024-10-07T14:15:59.493Z' },
 { id: 4, name: 'Bruno Mars', job: 'Singer-songwriter', last_updated: '2024-08-25T17:45:12.493Z' },
 { id: 1, name: 'Taylor Swift', job: 'Pop star', last_updated: '2024-09-20T10:12:43.493Z' }, // duplicate
```

```

 { id: 5, name: 'Billie Eilish', job: 'Singer-songwriter',
last_updated: '2024-09-10T09:30:12.493Z' },
 { id: 6, name: 'Katy Perry', job: 'Pop star', last_updated:
'2024-10-08T12:30:45.493Z' },
 { id: 2, name: 'Ed Sheeran', job: 'Singer-songwriter',
last_updated: '2024-10-05T08:30:59.493Z' }, // duplicate
 { id: 7, name: 'Lady Gaga', job: 'Pop star', last_updated:
'2024-09-15T14:45:30.493Z' },
 { id: 8, name: 'Rihanna', job: 'Pop star', last_updated: '2024-
10-01T11:50:22.493Z' },
 { id: 3, name: 'Adele', job: 'Singer-songwriter', last_updated:
'2024-10-07T14:15:59.493Z' }, // duplicate
 // { id: 9, name: 'Tom Hanks', job: 'Actor', last_updated: '2024-
10-17T13:58:31.493Z' },
 // { id: 0, name: 'Madonna', job: 'Pop star', last_updated:
'2024-10-17T17:11:38.493Z' },
 // { id: 15, name: 'Bob Dylan', job: 'Folk singer', last_updated:
'2024-09-24T08:03:16.493Z' },
 // { id: 10, name: 'Harry Nilsson', job: 'Singer-songwriter',
last_updated: '2020-10-17T17:11:38.493Z' },
 // { id: 11, name: 'Kylie Minogue', job: 'Pop star',
last_updated: '2024-10-24T08:03:16.493Z' },
]
}

```

4. Add a Split Out node to the canvas and connect it to the Code node.
5. In the Split Out node, enter data in the **Fields To Split Out** field.

## Removing duplicates from the current input

1. Add a Remove Duplicates node to the canvas and connect it to the Split Out node. Choose **Remove items repeated within current input** as the **Action** to start.
2. Open the Remove Duplicates node and ensure that the **Operation** is set to **Remove Items Repeated Within Current Input**.
3. Choose **All fields** in the **Compare** field.
4. Select **Execute step** to run the Remove Duplicates node, removing duplicated data in the current input.

n8n removes the items that have the same data across all fields. Your output in table view should look like this:

<b>id</b>	<b>name</b>	<b>job</b>	<b>last_updated</b>
1	Taylor Swift	Pop star	2024-09-20T10:12:43.493Z
2	Ed Sheeran	Singer-songwriter	2024-10-05T08:30:59.493Z
3	Adele	Singer-songwriter	2024-10-07T14:15:59.493Z
4	Bruno Mars	Singer-songwriter	2024-08-25T17:45:12.493Z
5	Billie Eilish	Singer-songwriter	2024-09-10T09:30:12.493Z
			2024-10-

6	Katy Perry	Pop star	08T12:30:45.493Z
7	Lady Gaga	Pop star	2024-09-15T14:45:30.493Z
8	Rihanna	Pop star	2024-10-01T11:50:22.493Z

5. Open the Remove Duplicates node again and change the **Compare** parameter to **Selected Fields**.
6. In the **Fields To Compare** field, enter job.
7. Select **Execute step** to run the Remove Duplicates node, removing duplicated data in the current input.

n8n removes the items in the current input that have the same job data. Your output in table view should look like this:

<b>id</b>	<b>name</b>	<b>job</b>	<b>last_updated</b>
1	Taylor Swift	Pop star	2024-09-20T10:12:43.493Z
2	Ed Sheeran	Singer-songwriter	2024-10-05T08:30:59.493Z

## Keep items where the value is new

1. Open the Remove Duplicates node and set the **Operation** to **Remove Items Processed in Previous Executions**.
2. Set the **Keep Items Where** parameter to **Value Is New**.
3. Set the **Value to Dedupe On** parameter to `{{ $json.name }}`.
4. On the canvas, select **Execute workflow** to run the workflow. Open the Remove Duplicates node to examine the results.

n8n compares the current input data to the items stored from previous executions. Since this is the first time running the Remove Duplicates node with this operation, n8n processes all data items and places them into the **Kept** output tab. The order of the items may be different than the order in the input data:

<b>id</b>	<b>name</b>	<b>job</b>	<b>last_updated</b>
1	Taylor Swift	Pop star	2024-09-20T10:12:43.493Z
1	Taylor Swift	Pop star	2024-09-20T10:12:43.493Z
2	Ed Sheeran	Singer-songwriter	2024-10-05T08:30:59.493Z
2	Ed Sheeran	Singer-songwriter	2024-10-05T08:30:59.493Z
3	Adele	Singer-songwriter	2024-10-07T14:15:59.493Z
3	Adele	Singer-songwriter	2024-10-07T14:15:59.493Z
4	Bruno Mars	Singer-songwriter	2024-08-25T17:45:12.493Z
5	Billie Eilish	Singer-songwriter	2024-09-10T09:30:12.493Z

6	Katy Perry	Pop star	2024-10-08T12:30:45.493Z
7	Lady Gaga	Pop star	2024-09-15T14:45:30.493Z
8	Rihanna	Pop star	2024-10-01T11:50:22.493Z

5. Open the Code node and uncomment (remove the // from) the line for "Tom Hanks."
6. On the canvas, select **Execute workflow** again. Open the Remove Duplicates node again to examine the results.

n8n compares the current input data to the items stored from previous executions. This time, the **Kept** tab contains the one new record from the Code node:

<b>id</b>	<b>name</b>	<b>job</b>	<b>last_updated</b>
9	Tom Hanks	Actor	2024-10-17T13:58:31.493Z

The **Discarded** tab contains the items processed by the previous execution:

<b>id</b>	<b>name</b>	<b>job</b>	<b>last_updated</b>
1	Taylor Swift	Pop star	2024-09-20T10:12:43.493Z
1	Taylor Swift	Pop star	2024-09-20T10:12:43.493Z
2	Ed Sheeran	Singer-songwriter	2024-10-05T08:30:59.493Z
2	Ed Sheeran	Singer-songwriter	2024-10-05T08:30:59.493Z
3	Adele	Singer-songwriter	2024-10-07T14:15:59.493Z
3	Adele	Singer-songwriter	2024-10-07T14:15:59.493Z
4	Bruno Mars	Singer-songwriter	2024-08-25T17:45:12.493Z
5	Billie Eilish	Singer-songwriter	2024-09-10T09:30:12.493Z
6	Katy Perry	Pop star	2024-10-08T12:30:45.493Z
7	Lady Gaga	Pop star	2024-09-15T14:45:30.493Z
8	Rihanna	Pop star	2024-10-01T11:50:22.493Z

Before continuing, clear the duplication history to get ready for the next example:

7. Open the Remove Duplicates node and set the **Operation** to **Clear Deduplication History**.
8. Select **Execute step** to clear the current duplication history.

## Keep items where the value is higher than any previous value

1. Open the Remove Duplicates node and set the **Operation** to **Remove Items Processed in Previous Executions**.
2. Set the **Keep Items Where** parameter to **Value Is Higher than Any Previous Value**.
3. Set the **Value to Dedupe On** parameter to `{} $json.id {}`.
4. On the canvas, select **Execute workflow** to run the workflow. Open the Remove Duplicates node to examine the results.

n8n compares the current input data to the items stored from previous executions. Since this is the first time running the Remove Duplicates node after clearing the history, n8n processes all data items and places them into the **Kept** output tab. The order of the items may be different than the order in the input data:

<b>id</b>	<b>name</b>	<b>job</b>	<b>last_updated</b>
1	Taylor Swift	Pop star	2024-09-20T10:12:43.493Z
1	Taylor Swift	Pop star	2024-09-20T10:12:43.493Z
2	Ed Sheeran	Singer-songwriter	2024-10-05T08:30:59.493Z
2	Ed Sheeran	Singer-songwriter	2024-10-05T08:30:59.493Z
3	Adele	Singer-songwriter	2024-10-07T14:15:59.493Z
3	Adele	Singer-songwriter	2024-10-07T14:15:59.493Z
4	Bruno Mars	Singer-songwriter	2024-08-25T17:45:12.493Z
5	Billie Eilish	Singer-songwriter	2024-09-10T09:30:12.493Z
6	Katy Perry	Pop star	2024-10-08T12:30:45.493Z
7	Lady Gaga	Pop star	2024-09-15T14:45:30.493Z
8	Rihanna	Pop star	2024-10-01T11:50:22.493Z
9	Tom Hanks	Actor	2024-10-17T13:58:31.493Z

5. Open the Code node and uncomment (remove the // from) the lines for "Madonna" and "Bob Dylan."
6. On the canvas, select **Execute workflow** again. Open the Remove Duplicates node again to examine the results.

n8n compares the current input data to the items stored from previous executions. This time, the **Kept** tab contains a single entry for "Bob Dylan." n8n keeps this item because its id column value (15) is higher than any previous values (the previous maximum value was 9):

<b>id</b>	<b>name</b>	<b>job</b>	<b>last_updated</b>
-----------	-------------	------------	---------------------

15 Bob Dylan Folk singer 2024-09-24T08:03:16.493Z

---

The **Discarded** tab contains the 13 items with an **id** column value equal to or less than the previous maximum value (9). Even though it's new, this table includes the entry for "Madonna" because its **id** value isn't larger than the previous maximum value:

<b>id</b>	<b>name</b>	<b>job</b>	<b>last_updated</b>
0	Madonna	Pop star	2024-10-17T17:11:38.493Z
1	Taylor Swift	Pop star	2024-09-20T10:12:43.493Z
1	Taylor Swift	Pop star	2024-09-20T10:12:43.493Z
2	Ed Sheeran	Singer-songwriter	2024-10-05T08:30:59.493Z
2	Ed Sheeran	Singer-songwriter	2024-10-05T08:30:59.493Z
3	Adele	Singer-songwriter	2024-10-07T14:15:59.493Z
3	Adele	Singer-songwriter	2024-10-07T14:15:59.493Z
4	Bruno Mars	Singer-songwriter	2024-08-25T17:45:12.493Z
5	Billie Eilish	Singer-songwriter	2024-09-10T09:30:12.493Z
6	Katy Perry	Pop star	2024-10-08T12:30:45.493Z
7	Lady Gaga	Pop star	2024-09-15T14:45:30.493Z
8	Rihanna	Pop star	2024-10-01T11:50:22.493Z
9	Tom Hanks	Actor	2024-10-17T13:58:31.493Z

Before continuing, clear the duplication history to get ready for the next example:

7. Open the Remove Duplicates node and set the **Operation** to **Clear Deduplication History**.
8. Select **Execute step** to clear the current duplication history.

## Keep items where the value is a date later than any previous date

1. Open the Remove Duplicates node and set the **Operation** to **Remove Items Processed in Previous Executions**.
2. Set the **Keep Items Where** parameter to **Value Is a Date Later than Any Previous Date**.
3. Set the **Value to Dedupe On** parameter to `{{ $json.last_updated }}`.
4. On the canvas, select **Execute workflow** to run the workflow.  
Open the Remove Duplicates node to examine the results.

n8n compares the current input data to the items stored from previous executions. Since this is the first time running the Remove Duplicates node after clearing the history, n8n processes all data items and places them into the **Kept** output tab. The order of the items may be different than the order in the input data:

<b>id</b>	<b>name</b>	<b>job</b>	<b>last_updated</b>
0	Madonna	Pop star	2024-10-17T17:11:38.493Z
1	Taylor Swift	Pop star	2024-09-20T10:12:43.493Z
1	Taylor Swift	Pop star	2024-09-20T10:12:43.493Z
2	Ed Sheeran	Singer-songwriter	2024-10-05T08:30:59.493Z
2	Ed Sheeran	Singer-songwriter	2024-10-05T08:30:59.493Z
3	Adele	Singer-songwriter	2024-10-07T14:15:59.493Z
3	Adele	Singer-songwriter	2024-10-07T14:15:59.493Z
4	Bruno Mars	Singer-songwriter	2024-08-25T17:45:12.493Z
5	Billie Eilish	Singer-songwriter	2024-09-10T09:30:12.493Z
6	Katy Perry	Pop star	2024-10-08T12:30:45.493Z
7	Lady Gaga	Pop star	2024-09-15T14:45:30.493Z
8	Rihanna	Pop star	2024-10-01T11:50:22.493Z
9	Tom Hanks	Actor	2024-10-17T13:58:31.493Z
15	Bob Dylan	Folk singer	2024-09-24T08:03:16.493Z

5. Open the Code node and uncomment (remove the // from) the lines for "Harry Nilsson" and "Kylie Minogue."
6. On the canvas, select **Execute workflow** again. Open the Remove Duplicates node again to examine the results.

n8n compares the current input data to the items stored from previous executions. This time, the **Kept** tab contains a single entry for "Kylie Minogue." n8n keeps this item because its `last_updated` column value (2024-10-24T08:03:16.493Z) is later than any previous values (the previous latest date was 2024-10-17T17:11:38.493Z):

<b>id</b>	<b>name</b>	<b>job</b>	<b>last_updated</b>
11	Kylie Minogue	Pop star	2024-10-24T08:03:16.493Z

The **Discarded** tab contains the 15 items with a `last_updated` column value equal to or earlier than the previous latest date (2024-10-17T17:11:38.493Z). Even though it's new, this table includes the entry

for “Harry Nilsson” because its `last_updated` value isn’t later than the previous maximum value:

<b>id</b>	<b>name</b>	<b>job</b>	<b>last_updated</b>
10	Harry Nilsson	Singer-songwriter	2020-10-17T17:11:38.493Z
0	Madonna	Pop star	2024-10-17T17:11:38.493Z
1	Taylor Swift	Pop star	2024-09-20T10:12:43.493Z
1	Taylor Swift	Pop star	2024-09-20T10:12:43.493Z
2	Ed Sheeran	Singer-songwriter	2024-10-05T08:30:59.493Z
2	Ed Sheeran	Singer-songwriter	2024-10-05T08:30:59.493Z
3	Adele	Singer-songwriter	2024-10-07T14:15:59.493Z
3	Adele	Singer-songwriter	2024-10-07T14:15:59.493Z
4	Bruno Mars	Singer-songwriter	2024-08-25T17:45:12.493Z
5	Billie Eilish	Singer-songwriter	2024-09-10T09:30:12.493Z
6	Katy Perry	Pop star	2024-10-08T12:30:45.493Z
7	Lady Gaga	Pop star	2024-09-15T14:45:30.493Z
8	Rihanna	Pop star	2024-10-01T11:50:22.493Z
9	Tom Hanks	Actor	2024-10-17T13:58:31.493Z
15	Bob Dylan	Folk singer	2024-09-24T08:03:16.493Z

## Rename Keys

Use the Rename Keys node to rename the keys of a key-value pair in n8n.

### Node parameters

You can rename one or multiple keys using the Rename Keys node. Select the **Add new key** button to rename a key.

For each key, enter the:

- **Current Key Name:** The current name of the key you want to rename.
- **New Key Name:** The new name you want to assign to the key.

## Node options

Choose whether to use a **Regex** regular expression to identify keys to rename. To use this option, you must also enter:

- The **Regular Expression** you'd like to use.
- **Replace With**: Enter the new name you want to assign to the key(s) that match the **Regular Expression**.
- You can also choose these Regex-specific options:
  - **Case Insensitive**: Set whether the regular expression should match case (turned off) or be case insensitive (turned on).
  - **Max Depth**: Enter the maximum depth to replace keys, using -1 for unlimited and 0 for top-level only.

## Templates and examples

---

## Respond to Chat node

Use the Respond to Chat node in correspondence with the [Chat Trigger](#) node to send a response into the chat and optionally wait for a response from the user. This allows you to have multiple chat interactions within a single execution and enables human-in-the-loop use cases in the chat. The Respond to Chat node also works as a tool for AI Agents.

## Node parameters

### Message

The message to send to the chat.

### Wait for User Reply

Set whether the workflow execution should wait for a response from the user (enabled) or continue immediately after sending the message (disabled).

## Node options

### Add Memory Input Connection

Choose whether you want to commit the messages from the Respond to Chat node to a connected memory. Using a shared memory between an agent or chain [root node](#) and the Respond to Chat node attaches the same session key to these messages and lets you capture the full message history.

### Limit Wait Time

When you enable **Wait for User Reply**, this option decides whether the workflow automatically resumes execution after a specific limit (enabled) or not (disabled).

## Related resources

-8<- “\_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

## Common issues

- The Respond to Chat node does not work when used as a tool of a subagent.
- The Respond to Chat node does not work when used in a subworkflow. This includes usage in a subworkflow that's being used as a tool for an AI Agent.

For common questions or issues and suggested solutions with the Chat Trigger node, refer to [Common Chat Trigger Node Issues](#).

---

## Respond to Webhook

Use the Respond to Webhook node to control the response to incoming webhooks. This node works with the [Webhook](#) node.

### How to use Respond to Webhook

To use the Respond to Webhook node:

1. Add a [Webhook](#) node as the trigger node for the workflow.
2. In the Webhook node, set **Respond to Using ‘Respond to Webhook’ node**.
3. Add the Respond to Webhook node anywhere in your workflow. If you want it to return data from other nodes, place it after those nodes.

### Node parameters

Configure the node behavior using these parameters.

#### Respond With

Choose what data to send in the webhook response.

- **All Incoming Items**: Respond with all the JSON items from the input.
- **Binary File**: Respond with a binary file defined in **Response Data Source**.
- **First Incoming Item**: Respond with the first incoming item's JSON.
- **JSON**: Respond with a JSON object defined in **Response Body**.
- **JWT Token**: Respond with a JSON Web Token (JWT).

- **No Data:** No response payload.
- **Redirect:** Redirect to a URL set in **Redirect URL**.
- **Text:** Respond with text set in **Response Body**. This sends HTML by default (Content-Type: text/html).

## Node options

Select **Add Option** to view and set the options.

- **Response Code:** Set the [response code](#) to use.
- **Response Headers:** Define the response headers to send.
- **Put Response in Field:** Available when you respond with **All Incoming Items** or **First Incoming Item**. Set the field name for the field containing the response data.
- **Enable Streaming:** When enabled, sends the data back to the user using streaming. Requires a trigger configured with the **Response mode Streaming**.

## How n8n secures HTML responses

Starting with n8n version 1.103.0, n8n automatically wraps HTML responses to webhooks in <iframe> tags. This is a security mechanism to protect the instance users.

This has the following implications:

- HTML renders in a sandboxed iframe instead of directly in the parent document.
- JavaScript code that attempts to access the top-level window or local storage will fail.
- Authentication headers aren't available in the sandboxed iframe (for example, basic auth). You need to use an alternative approach, like embedding a short-lived access token within the HTML.
- Relative URLs (for example, <form action="/">) won't work. Use absolute URLs instead.

## Templates and examples

## Workflow behavior

When using the Respond to Webhook node, workflows behave as follows:

- The workflow finishes without executing the Respond to Webhook node: it returns a standard message with a 200 status.
- The workflow errors before the first Respond to Webhook node executes: the workflow returns an error message with a 500 status.
- A second Respond to Webhook node executes after the first one: the workflow ignores it.
- A Respond to Webhook node executes but there was no webhook: the workflow ignores the Respond to Webhook node.

## Output the response sent to the webhook

By default, the Respond to Webhook node has a single output branch that contains the node's input data.

You can optionally enable a second output branch containing the response sent to the webhook. To enable this secondary output, open the Respond to Webhook node on the canvas and select the **Settings** tab. Activate the **Enable Response Branch** option.

The node will now have two outputs:

- **Input Data:** The original output, passing on the node's input.
- **Response:** The response object sent to the webhook.

## Return more than one data item (deprecated)

The Respond to Webhook node runs once, using the first incoming data item. This includes when using [expressions](#). You can't force looping using the Loop node: the workflow will run, but the webhook response will still only contain the results of the first execution.

If you need to return more than one data item, choose one of these options:

- Instead of using the Respond to Webhook node, use the **When Last Node Finishes** option in **Respond** in the Webhook node. Use this when you want to return the final data that the workflow outputs.
- Use the [Aggregate](#) node to turn multiple items into a single item before passing the data to the Respond to Webhook node. Set **Aggregate to All Item Data (Into a Single List)**.

---

## RSS Read

Use the RSS Read node to read data from RSS feeds published on the internet.

### Node parameters

- **URL:** Enter the URL for the RSS publication you want to read.

### Node options

- **Ignore SSL Issues:** Choose whether n8n should ignore SSL/TLS verification (turned on) or not (turned off).

### Templates and examples

### Related resources

n8n provides a trigger node for RSS Read. You can find the trigger node docs [here](#).

---

## RSS Feed Trigger node

The RSS Feed Trigger node allows you to start an n8n workflow when a new RSS feed item has been published.

On this page, you'll find a list of operations the RSS Feed Trigger node supports, and links to more resources.

### Node parameters

- **Poll Times:** Select a poll **Mode** to set how often to trigger the poll. Your **Mode** selection will add or remove relevant fields. Refer to the sections below to configure the parameters for each mode type.
- **Feed URL:** Enter the URL of the RSS feed to poll.

-8<- “\_snippets/integrations/builtin/poll-modes.md”

### Templates and examples

### Related resources

n8n provides an app node for RSS Feeds. You can find the node docs [here](#).

---

## Schedule Trigger node

Use the Schedule Trigger node to run workflows at fixed intervals and times. This works in a similar way to the Cron software utility in Unix-like systems.

-8<- “\_snippets/integrations/builtin/core-nodes/schedule/timezone-settings.md”

### Node parameters

Add **Trigger Rules** to determine when the trigger should run.

Use the **Trigger Interval** to select the time interval unit of measure to schedule the trigger for. All other parameters depend on the interval you select. Choose from:

- [Seconds trigger interval](#)
- [Minutes trigger interval](#)
- [Hours trigger interval](#)
- [Days trigger interval](#)
- [Weeks trigger interval](#)

- [Months trigger interval](#)
- [Custom \(Cron\) interval](#)

You can add multiple **Trigger Rules** to run the node on different schedules.

Refer to the sections below for more detail on configuring each **Trigger Interval**. Refer to [Templates and examples](#) for further examples.

## Seconds trigger interval

- **Seconds Between Triggers:** Enter the number of seconds between each workflow trigger. For example, if you enter 30 here, the trigger will run every 30 seconds.

## Minutes trigger interval

- **Minutes Between Triggers:** Enter the number of minutes between each workflow trigger. For example, if you enter 5 here, the trigger will run every 5 minutes.

## Hours trigger interval

- **Hours Between Triggers:** Enter the number of hours between each workflow trigger.
- **Trigger at Minute:** Enter the minute past the hour to trigger the node when it runs, from 0 to 59.

For example, if you enter 6 **Hours Between Triggers** and 30 **Trigger at Minute**, the node will run every six hours at 30 minutes past the hour.

## Days trigger interval

- **Days Between Triggers:** Enter the number of days between each workflow trigger.
- **Trigger at Hour:** Select the hour of the day to trigger the node.
- **Trigger at Minute:** Enter the minute past the hour to trigger the node when it runs, from 0 to 59.

For example, if you enter 2 **Days Between Triggers**, 9am for **Trigger at Hour**, and 15 **Trigger at Minute**, the node will run every two days at 9:15am.

## Weeks trigger interval

- **Weeks Between Triggers:** Enter the number of weeks between each workflow trigger.
- **Trigger on Weekdays:** Select the day(s) of the week you want to trigger the node.
- **Trigger at Hour:** Select the hour of the day to trigger the node.
- **Trigger at Minute:** Enter the minute past the hour to trigger the node when it runs, from 0 to 59.

For example, if you enter **2 Weeks Between Triggers**, **Monday** for **Trigger on Weekdays**, **3pm** for **Trigger at Hour**, and **30 Trigger at Minute**, the node will run every two weeks on Monday at 3:30 PM.

## Months trigger interval

- **Months Between Triggers:** Enter the number of months between each workflow trigger.
- **Trigger at Day of Month:** Enter the day of the month the day should trigger at, from 1 to 31. If a month doesn't have this day, the node won't trigger. For example, if you enter 30 here, the node won't trigger in February.
- **Trigger at Hour:** Select the hour of the day to trigger the node.
- **Trigger at Minute:** Enter the minute past the hour to trigger the node when it runs, from 0 to 59.

For example, if you enter **3 Months Between Triggers**, **28 Trigger at Day of Month**, **9am** for **Trigger at Hour**, and **0 Trigger at Minute**, the node will run each quarter on the 28th day of the month at 9:00 AM.

## Custom (Cron) interval

Enter a custom cron **Expression** to set the schedule for the trigger.

To generate a Cron expression, you can use [crontab guru](#). Paste the Cron expression that you generated using crontab guru in the **Expression** field in n8n.

## Examples

Type	Cron Expression	Description
Every X Seconds	*/10 * * * *	Every 10 seconds.
Every X Minutes	*/5 * * * *	Every 5 minutes.
Hourly	0 * * * *	Every hour on the hour.
Daily	0 6 * * *	At 6:00 AM every day.
Weekly	0 12 * * 1	At noon every Monday.
Monthly	0 0 1 * *	At midnight on the 1st of every month.
Every X Days	0 0 */3 * *	At midnight every 3rd day.
Only Weekdays	0 9 * * 1-5	At 9:00 AM Monday through Friday.
Custom Hourly Range	0 9-17 * * *	Every hour from 9:00 AM to 5:00 PM every day.
Quarterly	0 0 1 1,4,7,10 *	At midnight on the 1st of January, April, July, and October.

## Why there are six asterisks in the Cron expression

The sixth asterisk in the Cron expression represents seconds. Setting this is optional. The node will execute even if you don't set the value for seconds.

(*)	*	*	*	*	*
(second)	minute	hour	day of month	month	day of week(Sun-Sat)

## Templates and examples

### Common issues

For common questions or issues and suggested solutions, refer to [Common Issues](#).

---

## Schedule Trigger node common issues

Here are some common errors and issues with the [Schedule Trigger node](#) and steps to resolve or troubleshoot them.

### Invalid cron expression

This error occurs when you set **Trigger Interval** to **Custom (Cron)** and n8n doesn't understand your cron expression. This may mean that there is a mistake in your cron expression or that you're using an incompatible syntax.

To debug, check that the following:

- That your cron expression follows the syntax used in the [cron examples](#)
- That your cron expression (after removing the [seconds column](#)) validates on [crontab guru](#)

### Scheduled workflows run at the wrong time

If the Schedule Trigger node runs at the wrong time, it may mean that you need to adjust the time zone n8n uses.

#### Adjust the timezone globally

If you're using [n8n Cloud](#), follow the instructions on the [set the Cloud instance timezone](#) page to ensure that n8n executes in sync with your local time.

If you're [self hosting](#), set your global timezone using the [GENERIC\\_TIMEZONE environment variable](#).

## Adjust the timezone for an individual workflow

To set the timezone for an individual workflow:

1. Open the workflow on the canvas.
2. Select the [Image: three dots menu] **Three dots icon** in the upper-right corner.
3. Select **Settings**.
4. Change the **Timezone** setting.
5. Select **Save**.

## Variables not working as expected

While variables can be used in the scheduled trigger, their values only get evaluated when the workflow is published. After publishing the workflow, you can alter a variable's value in the settings but it won't change how often the workflow runs. To work around this, you must stop and then publish a new version of the workflow to apply the updated variable value.

## Changing the trigger interval

You can update the scheduled trigger interval at any time but it only gets updated when the workflow is published. If you change the trigger interval after the workflow is active, the changes won't take effect until you stop and then publish a new version of the workflow.

Also, the schedule begins from the time when you publish the workflow. For example, if you had originally set a schedule of every 1 hour and it should execute at 12:00, if you changed it to a 2 hour schedule and publish a new version of the workflow at 11:30, the next execution will be at 13:30, 2 hours from when you published it.

---

# Send Email

The Send Email node sends emails using an SMTP email server.

## Node parameters

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

Configure this node using the following parameters.

### Credential to connect with

Select or create an SMTP account credential for the node to use.

## Operation

The Send Email node supports the following operations:

- **Send:** Send an email.
- **Send and Wait for Response:** Send an email and wait for a response from the receiver. This operation pauses the workflow

execution until the user submits a response.

Choosing **Send and Wait for Response** will activate parameters and options as discussed in [waiting for a response](#).

## From Email

Enter the email address you want to send the email from. You can also include a name using this format: Name Name <email@example.com>, for example: Nathan Doe <nate@n8n.io>.

## To Email

Enter the email address you want to send the email to. You can also include a name using this format: Name Name <email@example.com>, for example: Nathan Doe <nate@n8n.io>. Use a comma to separate multiple email addresses: first@example.com, "Name" <second@example.com>.

## Subject

Enter the subject line for the email.

## Email Format

Select the format to send the email in. This parameter is available when using the **Send** operation. Choose from:

- **Text**: Send the email in plain-text format.
- **HTML**: Send the email in HTML format.
- **Both**: Send the email in both formats. If you choose this option, the email recipient's client will set which format to display.

## Node options

Use these **Options** to further refine the node's behavior.

### Append n8n Attribution

Set whether to include the phrase `This email was sent automatically with n8n` at the end of the email (turned on) or not (turned off).

### Attachments

Enter the name of the binary properties that contain data to add as an attachment. Some tips on using this option:

- Use the [Read/Write Files from Disk](#) node or the [HTTP Request](#) node to upload the file to your workflow.
- Add multiple attachments by entering a comma-separated list of binary properties.
- Reference embedded images or other content within the body of an email message, for example ``.

### CC Email

Enter an email address for the cc: field.

## BCC Email

Enter an email address for the bcc: field.

## Ignore SSL Issues

Set whether n8n should ignore failures with TLS/SSL certificate validation (turned on) or enforce them (turned off).

## Reply To

Enter an email address for the Reply To field.

## Waiting for a response

By choosing the **Send and Wait for a Response** operation, you can send an email message and pause the workflow execution until a person confirms the action or provides more information.

### Response Type

You can choose between the following types of waiting and approval actions:

- **Approval:** Users can approve or disapprove from within the message.
- **Free Text:** Users can submit a response with a form.
- **Custom Form:** Users can submit a response with a custom form.

Different options are available depending on which type you choose.

### Approval parameters and options

When using the Approval response type, the following options are available:

- **Type of Approval:** Whether to present only an approval button or both an approval and disapproval buttons.
- **Button Label:** The label for the approval or disapproval button. The default choice is Approve and Decline for approval and disapproval actions respectively.
- **Button Style:** The style (primary or secondary) for the button.

This mode also offers the following options:

- **Limit Wait Time:** Whether the workflow will automatically resume execution after a specified time limit. This can be an interval or a specific wall time.
- **Append n8n Attribution:** Set whether to include the phrase This email was sent automatically with n8n at the end of the email (turned on) or not (turned off).

### Free Text parameters and options

When using the Free Text response type, the following options are available:

- **Message Button Label:** The label to use for message button. The default choice is Respond.
- **Response Form Title:** The title of the form where users provide their response.
- **Response Form Description:** A description for the form where users provide their response.
- **Response Form Button Label:** The label for the button on the form to submit their response. The default choice is Submit.
- **Limit Wait Time:** Whether the workflow will automatically resume execution after a specified time limit. This can be an interval or a specific wall time.
- **Append n8n Attribution:** Set whether to include the phrase `This email was sent automatically with n8n` at the end of the email (turned on) or not (turned off).

## Custom Form parameters and options

When using the Custom Form response type, you build a form using the fields and options you want.

You can customize each form element with the settings outlined in the [n8n Form trigger's form elements](#). To add more fields, select the **Add Form Element** button.

The following options are also available:

- **Message Button Label:** The label to use for message button. The default choice is Respond.
- **Response Form Title:** The title of the form where users provide their response.
- **Response Form Description:** A description for the form where users provide their response.
- **Response Form Button Label:** The label for the button on the form to submit their response. The default choice is Submit.
- **Limit Wait Time:** Whether the workflow will automatically resume execution after a specified time limit. This can be an interval or a specific wall time.
- **Append n8n Attribution:** Set whether to include the phrase `This email was sent automatically with n8n` at the end of the email (turned on) or not (turned off).

## Templates and examples

---

## Sort

Use the Sort node to organize lists of items in a desired ordering, or generate a random selection.

### Node parameters

Configure this node using the **Type** parameter.

Use the dropdown to select how you want to input the sorting from these options.

## Simple

Performs an ascending or descending sort using the selected fields.

When you select this **Type**:

- Use the **Add Field To Sort By** button to input the **Field Name**.
- Select whether to use **Ascending** or **Descending** order.

### Simple options

When you select **Simple** as the **Type**, you have the option to **Disable Dot Notation**. By default, n8n enables dot notation to reference child fields in the format parent.child. Use this option to disable dot notation (turned on) or to continue using dot (turned off).

## Random

Creates a random order in the list.

## Code

Input custom JavaScript code to perform the sort operation. This is a good option if a simple sort won't meet your needs.

Enter your custom JavaScript code in the **Code** input field.

## Templates and examples

## Related resources

-8<- “\_snippets/integrations/builtin/core-nodes/data-transformation-actions/data-section-link.md”

---

# Split Out

Use the Split Out node to separate a single data item containing a list into multiple items. For example, a list of customers, and you want to split them so that you have an item for each customer.

## Node parameters

Configure this node using the following parameters.

### Field to Split Out

Enter the field containing the list you want to separate out into individual items.

If you're working with binary data inputs, use `$binary` in an expression to set the field to split out.

## Include

Select whether and how you want n8n to keep any other fields from the input data with each new individual item.

You can select:

- **No Other Fields:** No other fields will be included.
- **All Other Fields:** All other fields will be included.
- **Selected Other Fields:** Only the selected fields will be included.
  - **Fields to Include:** Enter a comma separated list of the fields you want to include.

## Node options

### Disable Dot Notation

By default, n8n enables dot notation to reference child fields in the format `parent.child`. Use this option to disable dot notation (turned on) or to continue using dot (turned off).

### Destination Field Name

Enter the field in the output where the split field contents should go.

### Include Binary

Choose whether to include binary data from the input in the new output (turned on) or not (turned off).

## Templates and examples

### Related resources

-8<- “\_snippets/integrations/builtin/core-nodes/data-transformation-actions/data-section-link.md”

---

## SSE Trigger node

Server-Sent Events (SSE) is a server push technology enabling a client to receive automatic updates from a server using HTTP connection. The SSE Trigger node is used to receive server-sent events.

## Node parameters

The SSE Trigger node has one parameter, the **URL**. Enter the URL from which to receive the server-sent events (SSE).

## Templates and examples

---

# SSH

The SSH node is useful for executing commands using the Secure Shell Protocol.

## Operations

- [Execute a command](#)
- [Download a file](#)
- [Upload a file](#)

### Execute Command

Configure this operation with these parameters:

- **Credential to connect with:** Select an existing or create a new [SSH credential](#) to connect with.
- **Command:** Enter the command to execute on the remote device.
- **Working Directory:** Enter the directory where n8n should execute the command.

### Download File

- **Credential to connect with:** Select an existing or create a new [SSH credential](#) to connect with.
- **Path:** Enter the path for the file you want to download. This path must include the file name. The downloaded file will use this file name. To use a different name, use the **File Name** option. Refer to [Download File options](#) for more information.
- **File Property:** Enter the name of the object property that holds the binary data you want to download.

#### Download File options

You can further configure this operation with the **File Name** option. Use this option to override the binary data file name to a name of your choice.

### Upload File

- **Credential to connect with:** Select an existing or create a new [SSH credential](#) to connect with.
- **Input Binary Field:** Enter the name of the input binary field that contains the file you want to upload.
- **Target Directory:** The directory to upload the file to. The name of the file is taken from the binary data file name. To enter a different name, use the **File Name** option. Refer to [Upload File options](#) for more information.

#### Upload File options

You can further configure this operation with the **File Name** option. Use this option to override the binary data file name to a name of your choice.

## Templates and examples

---

### Stop And Error

Use the Stop And Error node to display custom error messages, cause executions to fail under certain conditions, and send custom error information to error workflows.

### Operations

- Error Message
- Error Object

### Node parameters

Both operations include one node parameter, the **Error Type**. Use this parameter to select the type of error to throw. Choose between the two operations: **Error Message** and **Error Object**.

The other parameters depend on which operation you select.

#### Error Message parameters

The Error Message Error Type adds one parameter, the **Error Message** field. Enter the message you'd like to throw.

#### Error Object parameters

The Error Object Error Type adds one parameter, the **Error Object**. Enter a JSON object that contains the error properties you'd like to throw.

## Templates and examples

### Related resources

You can use the Stop And Error node with the [Error trigger node](#).

Read more about [Error workflows](#) in n8n workflows.

---

## Summarize

Use the Summarize node to aggregate items together, in a manner similar to Excel pivot tables.

## Node parameters

### Fields to Summarize

Use these fields to define how you want to summarize your input data.

- **Aggregation:** Select the aggregation method to use on a given field. Options include:
  - **Append:** Append
    - If you select this option, decide whether you want to **Include Empty Values** or not.
  - **Average:** Calculate the numeric average of your input data.
  - **Concatenate:** Combine together values in your input data.
    - If you select this option, decide whether you want to **Include Empty Values** or not.
  - **Separator:** Select the separator you want to insert between concatenated values.
  - **Count:** Count the total number of values in your input data.
  - **Count Unique:** Count the number of unique values in your input data.
  - **Max:** Find the highest numeric value in your input data.
  - **Min:** Find the lowest numeric value in your input data.
  - **Sum:** Add together the numeric values in your input data.
- **Field:** Enter the name of the field you want to perform the aggregation on.

### Fields to Split By

Enter the name of the input fields that you want to split the summary by (similar to a group by statement). This allows you to get separate summaries based on values in other fields.

For example, if our input data contains columns for Sales Rep and Deal Amount and we're performing a **Sum** on the Deal Amount field, we could split by Sales Rep to get a **Sum** total for each Sales Rep.

To enter multiple fields to split by, enter a comma-separated list.

## Node options

### Continue if Field Not Found

By default, if a **Field to Summarize** isn't in any items, the node throws an error. Use this option to continue and return a single empty item (turned on) instead or keep the default error behavior (turned off).

### Disable Dot Notation

By default, n8n enables dot notation to reference child fields in the format parent.child. Use this option to disable dot notation (turned on) or to continue using dot (turned off).

## Output Format

Select the format for your output format. This option is recommended if you're using **Fields to Split By**

- **Each Split in a Separate Item:** Use this option to generate a separate output item for each split out field.
- **All Splits in a Single Item:** Use this option to generate a single item that lists the split out fields.

## Ignore items without valid fields to group by

Set whether to ignore input items that don't contain the **Fields to Split By** (turned on) or not (turned off).

## Templates and examples

## Related resources

-8<- “\_snippets/integrations/builtin/core-nodes/data-transformation-actions/data-section-link.md”

---

## Switch

Use the Switch node to route a workflow conditionally based on comparison operations. It's similar to the IF node, but supports multiple output routes.

## Node parameters

Select the **Mode** the node should use:

- **Rules:** Select this mode to build a matching rule for each output.
- **Expression:** Select this mode to write an expression to return the output index programmatically.

Node configuration depends on the **Mode** you select.

## Rules

To configure the node with this operation, use these parameters:

- Create **Routing Rules** to define comparison conditions.
  - Use the data type dropdown to select the data type and comparison operation type for your condition. For example, to create a rules for dates after a particular date, select **Date & Time > is after**.
  - The fields and values to enter into the condition change based on the data type and comparison you select. Refer to Available data type comparisons for a full list of all comparisons by data

type.

- **Rename Output:** Turn this control on to rename the output field to put matching data into. Enter your desired **Output Name**.

Select **Add Routing Rule** to add more rules.

### Rule options

You can further configure the node with this operation using these **Options**:

- **Fallback Output:** Choose how to route the workflow when an item doesn't match any of the rules or conditions.
  - **None:** Ignore the item. This is the default behavior.
  - **Extra Output:** Send items to an extra, separate output.
  - **Output 0:** Send items to the same output as those matching the first rule.
- **Ignore Case:** Set whether to ignore letter case when evaluating conditions (turned on) or enforce letter case (turned off).
- **Less Strict Type Validation:** Set whether you want n8n to attempt to convert value types based on the operator you choose (turned on) or not (turned off).
- **Send data to all matching outputs:** Set whether to send data to all outputs meeting conditions (turned on) or whether to send the data to the first output matching the conditions (turned off).

### Expression

To configure the node with this operation, use these parameters:

- **Number of Outputs:** Set how many outputs the node should have.
- **Output Index:** Create an expression to calculate which input item should be routed to which output. The expression must return a number.

## Templates and examples

### Related resources

Refer to [Splitting with conditionals](#) for more information on using conditionals to create complex logic in n8n.

-8<- “\_snippets/integrations/builtin/core-nodes/data-types.md”

---

## TOTP

The TOTP node provides a way to generate a TOTP (time-based one-time password).

### Node parameters

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

Configure this node with these parameters.

### Credential to connect with

Select or create a [TOTP credential](#) for the node to use.

### Operation

**Generate Secret** is the only operation currently supported.

## Node options

Use these **Options** to further configure the node.

### Algorithm

Select the HMAC hashing algorithm to use. Default is SHA1.

### Digits

Enter the number of digits in the generated code. Default is 6.

### Period

Enter how many seconds the TOTP is valid for. Default is 30.

## Templates and examples

---

## Wait

Use the Wait node pause your workflow's execution. When the workflow pauses it offloads the execution data to the database. When the resume condition is met, the workflow reloads the data and the execution continues.

## Operations

The Wait node can **Resume** on the following conditions:

- **After Time Interval**: The node waits for a certain amount of time.
- **At Specified Time**: The node waits until a specific time.
- **On Webhook Call**: The node waits until it receives an HTTP call.
- **On Form Submitted**: The node waits until it receives a form submission.

Refer to the more detailed sections below for more detailed instructions.

### After Time Interval

Wait for a certain amount of time.

This parameter includes two more fields:

- **Wait Amount:** Enter the amount of time to wait.
- **Wait Unit:** Select the unit of measure for the **Wait Amount**. Choose from:
  - **Seconds**
  - **Minutes**
  - **Hours**
  - **Days**

Refer to [Time-based operations](#) for more detail on how these intervals work and the timezone used.

## At Specified Time

Wait until a specific date and time to continue. Use the date and time picker to set the **Date and Time**.

Refer to [Time-based operations](#) for more detail on the timezone used.

## On Webhook Call

This parameter enables your workflows to resume when the Wait node receives an HTTP call.

The webhook URL that resumes the execution when called is generated at runtime. The Wait node provides the `$execution.resumeUrl` variable so that you can reference and send the yet-to-be-generated URL wherever needed, for example to a third-party service or in an email.

When the workflow executes, the Wait node generates the resume URL and the webhook(s) in your workflow using the `$execution.resumeUrl`. This generated URL is unique to each execution, so your workflow can contain multiple Wait nodes and as the webhook URL is called it will resume each Wait node sequentially.

For this **Resume** style, set more parameters listed below.

### Authentication

Select if and how incoming resume-webhook-requests to `$execution.resumeUrl` should be authenticated. Options include:

- **Basic Auth:** Use basic authentication. Select or enter a new **Credential for Basic Auth** to use.
- **Header Auth:** Use header authentication. Select or enter a new **Credential for Header Auth** to use.
- **JWT Auth:** Use JWT authentication. Select or enter a new **Credential for JWT Auth** to use.
- **None:** Don't use authentication.

### HTTP Method

Select the HTTP method the webhook should use. Refer to the [Webhook node | HTTP Method documentation](#) for more information.

## Response Code

Enter the Response Code the webhook should return. You can use common codes or enter a custom code.

## Respond

Set when and how to respond to the webhook from these options:

- **Immediately:** Respond as soon as the node executes.
- **When Last Node Finishes:** Return the response code and the data output from the last node executed in the workflow. If you select this option, also set:
  - **Response Data:** Select what data should be returned and what format to use. Options include:
    - **All Entries:** Returns all the entries of the last node in an array.
    - **First Entry JSON:** Return the JSON data of the first entry of the last node in a JSON object.
    - **First Entry Binary:** Return the binary data of the first entry of the last node in a binary file.
    - **No Response Body:** Return with no body.
- **Using 'Respond to Webhook' Node:** Respond as defined in the Respond to Webhook node.

## Limit Wait Time

Set whether the workflow will automatically resume execution after a specific limit type (turned on) or not (turned off). If turned on, also set:

- **Limit Type:** Select what type of limit to enforce from these options:
  - **After Time Interval:** Wait for a certain amount of time.
    - Enter the limit's **Amount** of time.
    - Select the limit's **Unit** of time.
  - **At Specified Time:** Wait until a specific date and time to resume.
    - **Max Date and Time:** Use the date and time picker to set the specified time the node should resume.

## On Webhook Call options

- **Binary Property:** Enter the name of the binary property to write the data of the received file to. This option's only relevant if binary data is received.
- **Ignore Bots:** Set whether to ignore requests from bots like link previewers and web crawlers (turned on) or not (turned off).
- **IP(s) Whitelist:** Enter IP addresses here to limit who (or what) can invoke the webhook URL. Enter a comma-separated list of allowed IP addresses. Access from IPs outside the whitelist throws a 403 error. If left blank, all IP addresses can invoke the webhook URL.
- **No Response Body:** Set whether n8n should send a body in the response (turned off) or prevent n8n from sending a body in the response (turned on).
- **Raw Body:** Set whether to return the body in a raw format like JSON or XML (turned on) or not (turned off).

- **Response Data:** Enter any custom data you want to send in the response.
- **Response Headers:** Send more headers in the webhook response. Refer to [MDN Web Docs | Response header](#) to learn more about response headers.
- **Webhook Suffix:** Enter a suffix to append to the resume URL. This is useful for creating unique webhook URLs for each Wait node when a workflow contains multiple Wait nodes. Note that the generated \$resumeWebhookUrl won't automatically include this suffix, you must manually append it to the webhook URL before exposing it.

## On Webhook Call limitations

There are some limitations to keep in mind when using On Webhook Call:

- Partial executions of your workflow changes the \$resumeWebhookUrl, so be sure that the node sending this URL to your desired third-party runs in the same execution as the Wait node.

## On Form Submitted

Wait for a form submission before continuing. Set up these parameters:

### Form Title

Enter the title to display at the top of the form.

### Form Description

Enter a form description to display beneath the title. This description can help prompt the user on how to complete the form.

### Form Fields

Set up each field you want to appear on your form using these parameters:

- **Field Label:** Enter the field label you want to appear in the form.
- **Field Type:** Select the type of field to display in the form. Choose from:
  - **Date**
  - **Dropdown List:** Enter each dropdown options in the **Field Options**.
    - **Multiple Choice:** Select whether the user can select a single dropdown option (turned off) or multiple dropdown options (turned on)
  - **Number**
  - **Password**
  - **Text**
  - **Textarea**
- **Required Field:** Set whether the user must complete this field in order to submit the form (turned on) or if the user can submit the form without completing it (turned off).

### Respond When

Set when to respond to the form submission. Choose from:

- **Form Is Submitted:** Respond as soon as this node receives the form submission.
- **Workflow Finishes:** Respond when the last node of this workflow finishes.
- **Using 'Respond to Webhook' Node:** Respond when the Respond to Webhook node executes.

### Limit Wait Time

Set whether the workflow will automatically resume execution after a specific limit type (turned on) or not (turned off).

If turned on, also set:  
\* **Limit Type:** Select what type of limit to enforce from these options:  
\* **After Time Interval:** Wait for a certain amount of time.  
\* Enter the limit's **Amount** of time.  
\* Select the limit's **Unit** of time.  
\* **At Specified Time:** Wait until a specific date and time to resume.  
\* **Max Date and Time:** Use the date and time picker to set the specified time the node should resume.

### On Form Response options

- **Form Response:** Choose how and what you want the form to **Respond With** from these options:
  - **Form Submitted Text:** The form displays whatever text is entered in **Text to Show** after a user fills out the form. Use this option if you want to display a confirmation message.
  - **Redirect URL:** The form will redirect the user to the **URL to Redirect to** after they fill out the form. This must be a valid URL.
- **Webhook Suffix:** Enter a suffix to append to the resume URL. This is useful for creating unique webhook URLs for each Wait node when a workflow contains multiple Wait nodes. Note that the generated \$resumeWebhookUrl won't automatically include this suffix, you must manually append it to the webhook URL before exposing it.

## Templates and examples

### Time-based operations

For the time-based resume operations, note that:

- For wait times less than 65 seconds, the workflow doesn't offload execution data to the database. Instead, the process continues to run and the execution resumes after the specified interval passes.
- The n8n server time is always used regardless of the timezone setting. Workflow timezone settings, and any changes made to them, don't affect the Wait node interval or specified time.

---

## Webhook node

Use the Webhook node to create [webhooks](#), which can receive data from apps and services when an event occurs. It's a trigger node, which means it can start an n8n workflow. This allows services to connect to n8n and run a workflow.

You can use the Webhook node as a trigger for a workflow when you want to receive data and run a workflow based on the data. The Webhook node also supports returning the data generated at the end of a workflow. This makes it useful for building a workflow to process data and return the results, like an API endpoint.

The webhook allows you to trigger workflows from services that don't have a dedicated app trigger node.

## Workflow development process

n8n provides different **Webhook URLs** for testing and production. The testing URL includes an option to **Listen for test event**. Refer to [Workflow development](#) for more information on building, testing, and shifting your Webhook node to production.

## Node parameters

Use these parameters to configure your node.

### Webhook URLs

The Webhook node has two **Webhook URLs**: test and production. n8n displays the URLs at the top of the node panel.

Select **Test URL** or **Production URL** to toggle which URL n8n displays.

[Image: Sample Webhook URLs in the Webhook node's Parameters tab display a Test URL and Production URL]  
Sample Webhook URLs in the Webhook node's Parameters tab

- **Test:** n8n registers a test webhook when you select **Listen for Test Event** or **Execute workflow**, if the workflow isn't active. When you call the webhook URL, n8n displays the data in the workflow.
- **Production:** n8n registers a production webhook when you publish the workflow. When using the production URL, n8n doesn't display the data in the workflow. You can still view workflow data for a production execution: select the **Executions** tab in the workflow, then select the workflow execution you want to view.

### HTTP Method

The Webhook node supports standard [HTTP Request Methods](#):

- DELETE
- GET
- HEAD
- PATCH
- POST

- PUT

## Path

By default, this field contains a randomly generated webhook URL path, to avoid conflicts with other webhook nodes.

You can manually specify a URL path, including adding route parameters. For example, you may need to do this if you use n8n to prototype an API and want consistent endpoint URLs.

The **Path** field can take the following formats:

- `/:variable`
- `/path/:variable`
- `/:variable/path`
- `/:variable1/path/:variable2`
- `/:variable1/:variable2`

## Supported authentication methods

You can require authentication for any service calling your webhook URL. Choose from these authentication methods:

- Basic auth
- Header auth
- JWT auth
- None

Refer to [Webhook credentials](#) for more information on setting up each credential type.

## Respond

- **Immediately:** The Webhook node returns the response code and the message **Workflow got started**.
- **When Last Node Finishes:** The Webhook node returns the response code and the data output from the last node executed in the workflow.
- **Using 'Respond to Webhook' Node:** The Webhook node responds as defined in the [Respond to Webhook](#) node.
- **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).

## Response Code

Customize the [HTTP response code](#) that the Webhook node returns upon successful execution. Select from common response codes or create a custom code.

## Response Data

Choose what data to include in the response body:

- **All Entries:** The Webhook returns all the entries of the last node in an array.

- **First Entry JSON:** The Webhook returns the JSON data of the first entry of the last node in a JSON object.
- **First Entry Binary:** The Webhook returns the binary data of the first entry of the last node in a binary file.
- **No Response Body:** The Webhook returns without a body.

Applies only to **Respond > When Last Node Finishes**.

## Node options

Select **Add Option** to view more configuration options. The available options depend on your node parameters. Refer to the table for option availability.

- **Allowed Origins (CORS):** Set the permitted cross-origin domains. Enter a comma-separated list of URLs allowed for cross-origin non-preflight requests. Use \* (default) to allow all origins.
- **Binary Property:** Enabling this setting allows the Webhook node to receive binary data, such as an image or audio file. Enter the name of the binary property to write the data of the received file to.
- **Ignore Bots:** Ignore requests from bots like link previewers and web crawlers.
- **IP(s) Whitelist:** Enable this to limit who (or what) can invoke a Webhook trigger URL. Enter a comma-separated list of allowed IP addresses. Access from IP addresses outside the whitelist throws a 403 error. If left blank, all IP addresses can invoke the webhook trigger URL.
- **No Response Body:** Enable this to prevent n8n sending a body with the response.
- **Raw Body:** Specify that the Webhook node will receive data in a raw format, such as JSON or XML.
- **Response Content-Type:** Choose the format for the webhook body.
- **Response Data:** Send custom data with the response.
- **Response Headers:** Send extra headers in the Webhook response. Refer to [MDN Web Docs | Response header](#) to learn more about response headers.
- **Property Name:** by default, n8n returns all available data. You can choose to return a specific JSON key, so that n8n returns the value.

Option	Required node configuration
Allowed Origins (CORS)	Any
Binary Property	Either: HTTP Method > POST HTTP Method > PATCH HTTP Method > PUT
Ignore Bots	Any
IP(s) Whitelist	Any
Property Name	Both: Respond > When Last Node Finishes Response Data > First Entry JSON

No	
Response Body	Respond > Immediately
Raw Body	Any
Response Code	Any except Respond > Using 'Respond to Webhook' Node
Response Content-Type	Both: Respond > When Last Node Finishes Response Data > First Entry JSON
Response Data	Respond > Immediately
Response Headers	Any

---

## How n8n secures HTML responses

Starting with n8n version 1.103.0, n8n automatically wraps HTML responses to webhooks in `<iFrame>` tags. This is a security mechanism to protect the instance users.

This has the following implications:

- HTML renders in a sandboxed iframe instead of directly in the parent document.
- JavaScript code that attempts to access the top-level window or local storage will fail.
- Authentication headers aren't available in the sandboxed iframe (for example, basic auth). You need to use an alternative approach, like embedding a short-lived access token within the HTML.
- Relative URLs (for example, `<form action="/">`) won't work. Use absolute URLs instead.

## Templates and examples

### Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

---

## Workflow development

The [Webhook node](#) works a bit differently from other core nodes. n8n recommends following these processes for building, testing, and using your Webhook node in production.

n8n generates two **Webhook URLs** for each Webhook node: a **Test URL** and a **Production URL**.

## Build and test workflows

While building or testing a workflow, use the **Test** webhook URL.

Using a test webhook ensures that you can view the incoming data in the editor UI, which is useful for debugging. Select **Listen for test event** to register the webhook before sending the data to the test webhook. The test webhook stays active for 120 seconds.

When using the Webhook node on localhost on a self-hosted n8n instance, run n8n in tunnel mode:

- [npm with tunnel](#)
- [Docker with tunnel](#)

## Production workflows

When your workflow is ready, switch to using the **Production** webhook URL. You can then publish your workflow, and n8n runs it automatically when an external service calls the webhook URL.

When working with a Production webhook, ensure that you have saved and published the workflow. Data flowing through the webhook isn't visible in the editor UI with the production webhook.

Refer to [Create a workflow](#) for more information on publishing workflows.

---

## Common issues and questions

Here are some common issues and questions for the Webhook node and suggested solutions.

### Listen for multiple HTTP methods

By default, the Webhook node accepts calls that use a single method. For example, it can accept GET or POST requests, but not both. If you want to accept calls using multiple methods:

1. Open the node **Settings**.
2. Turn on **Allow Multiple HTTP Methods**.
3. Return to **Parameters**. By default, the node now accepts GET and POST calls. You can add other methods in the **HTTP Methods** field.

The Webhook node has an output for each method, so you can perform different actions depending on the method.

### Use the HTTP Request node to trigger the Webhook node

The HTTP Request node makes HTTP requests to the URL you specify.

1. Create a new workflow.
2. Add the HTTP Request node to the workflow.
3. Select a method from the **Request Method** dropdown list. For example, if you select GET as the **HTTP method** in your Webhook node, select GET as the request method in the HTTP Request

- node.
4. Copy the URL from the Webhook node, and paste it in the **URL** field in the HTTP Request node.
  5. If using the test URL for the webhook node: execute the workflow with the Webhook node.
  6. Execute the HTTP Request node.

## Use curl to trigger the Webhook node

You can use `curl` to make HTTP requests that trigger the Webhook node.

Make an HTTP request without any parameters:

```
curl --request GET
```

Make an HTTP request with a body parameter:

```
curl --request GET --data 'key=value'
```

Make an HTTP request with header parameter:

```
curl --request GET --header 'key=value'
```

Make an HTTP request to send a file:

```
curl --request GET --from 'key=@/path/to/file'
```

Replace `/path/to/file` with the path of the file you want to send.

## Send a response of type string

By default, the response format is JSON or an array. To send a response of type string:

1. Select **Response Mode > When Last Node Finishes**.
2. Select **Response Data > First Entry JSON**.
3. Select **Add Option > Property Name**.
4. Enter the name of the property that contains the response. This defaults to `data`.
5. Connect an Edit Fields node to the Webhook node.
6. In the Edit Fields node, select **Add Value > String**.
7. Enter the name of the property in the **Name** field. The name should match the property name from step 4.
8. Enter the string value in the **Value** field.
9. Toggle **Keep Only Set** to on (green).

When you call the Webhook, it sends the string response from the Edit Fields node.

## Test URL versus Production URL

n8n generates two **Webhook URLs** for each Webhook node: a **Test URL** and a **Production URL**.

While building or testing a workflow, use the **Test URL**. Once you're ready to use your Webhook URL in production, use the **Production URL**.

URL type	How to trigger	Listening duration	Data shown in editor UI?
Test URL	Select <b>Listen for test event</b> and trigger a test event from the source.	120 seconds	:white_check_mark:
Production URL	Publish the workflow	Until workflow is unpublished	:x:

Refer to [Workflow development](#) for more information.

## IP addresses in whitelist are failing to connect

If you're unable to connect from IP addresses in your IP whitelist, check if you are running n8n behind a reverse proxy.

If so, set the `N8N_PROXY_HOPS` [environment variable](#) to the number of reverse-proxies n8n is running behind.

## Only one webhook per path and method

n8n only permits registering one webhook for each path and HTTP method combination (for example, a GET request for /my-request). This avoids ambiguity over which webhook should receive requests.

If you receive a message that the path and method you chose are already in use, you can either:

- Unpublish the workflow with the conflicting webhook.
- Change the webhook path and/or method for one of the conflicting webhooks.

## Timeouts on n8n Cloud

n8n Cloud uses Cloudflare to protect against malicious traffic. If your webhook doesn't respond within 100 seconds, the incoming request will fail with a [524 status code](#).

Because of this, for long-running processes that might exceed this limit, you may need to introduce polling logic by configuring two separate webhooks:

- One webhook to start the long-running process and send an immediate response.
- A second webhook that you can call at intervals to query the status of the process and retrieve the result once it's complete.

# Workflow Trigger node

The Workflow Trigger node gets triggered when a workflow is updated or activated.

The Workflow Trigger node gets triggered for the workflow that it gets added to. You can use the Workflow Trigger node to trigger a workflow to notify the state of the workflow.

## Node parameters

The node includes a single parameter to identify the **Events** that should trigger it. Choose from these events:

- **Active Workflow Updated:** If you select this event, the node triggers when this workflow is updated.
- **Workflow Activated:** If you select this event, the node triggers when this workflow is activated.

You can select one or both of these events.

## Templates and examples

---

## XML

Use the XML node to convert data from and to XML.

## Node parameters

- **Mode:** The format the data should be converted from and to.
  - **JSON to XML:** Converts data from JSON to XML.
  - **XML to JSON:** Converts data from XML to JSON.
- **Property Name:** Enter the name of the property which contains the data to convert.

## Node options

These options are available regardless of the **Mode** you select:

- **Attribute Key:** Enter the prefix used to access the attributes. Default is \$.
- **Character Key:** Enter the prefix used to access the character content. Default is \_.

All other options depend on the selected **Mode**.

### JSON to XML options

These options only appear if you select **JSON to XML** as the **Mode**:

- **Allow Surrogate Chars:** Set whether to allow using characters

from the Unicode surrogate blocks (turned on) or not (turned off).

- **Cdata**: Set whether to wrap text nodes in `<![CDATA[ ... ]]>` instead of escaping when it's required (turned on) or not (turned off).
  - Turning this option on doesn't add `<![CDATA[ ... ]]>` if it's not required.
- **Headless**: Set whether to omit the XML header (turned on) or include it (turned off).
- **Root Name**: Enter the root element name to use.

## XML to JSON options

These options only appear if you select **XML to JSON** as the **Mode**:

- **Explicit Array**: Set whether to put child nodes in an array (turned on) or create an array only if there's more than one child node (turned off).
- **Explicit Root**: Set whether to get the root node in the resulting object (turned on) or not (turned off).
- **Ignore Attributes**: Set whether to ignore all XML attributes and only create text nodes (turned on) or not (turned off).
- **Merge Attributes**: Set whether to merge attributes and child elements as properties of the parent (turned on) or key attributes off a child attribute object (turned off). This option is ignored if **Ignore Attribute** is turned on.
- **Normalize**: Set whether to trim whitespaces inside the text nodes (turned on) or not to trim them (turned off).
- **Normalize Tags**: Set whether to normalize all tag names to lowercase (turned on) or keep tag names as-is (turned off).
- **Trim**: Set whether to trim the whitespace at the beginning and end of text nodes (turned on) or to leave the whitespace as-is (turned off).

## Templates and examples

---

## Actions library

This section provides information about n8n's Actions.

---

## Action Network node

Use the Action Network node to automate work in Action Network, and integrate Action Network with other applications. n8n has built-in support for a wide range of Action Network features, including creating, updating, and deleting events, people, tags, and signatures.

On this page, you'll find a list of operations the Action Network node supports, and links to more resources.

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

## Operations

- Attendance
  - Create
  - Get
  - Get All
- Event
  - Create
  - Get
  - Get All
- Person
  - Create
  - Get
  - Get All
  - Update
- Person Tag
  - Add
  - Remove
- Petition
  - Create
  - Get
  - Get All
  - Update
- Signature
  - Create
  - Get
  - Get All
  - Update
- Tag
  - Create
  - Get
  - Get All

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## ActiveCampaign node

Use the ActiveCampaign node to automate work in ActiveCampaign, and integrate ActiveCampaign with other applications. n8n has built-in support for a wide range of ActiveCampaign features, including creating, getting, updating, and deleting accounts, contact, orders, e-commerce customers, connections, lists, tags, and deals.

On this page, you'll find a list of operations the ActiveCampaign node supports and links to more resources.

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

## Operations

- Account
  - Create an account

- Delete an account
  - Get data of an account
  - Get data of all accounts
  - Update an account
- Account Contact
  - Create an association
  - Delete an association
  - Update an association
- Contact
  - Create a contact
  - Delete a contact
  - Get data of a contact
  - Get data of all contact
  - Update a contact
- Contact List
  - Add contact to a list
  - Remove contact from a list
- Contact Tag
  - Add a tag to a contact
  - Remove a tag from a contact
- Connection
  - Create a connection
  - Delete a connection
  - Get data of a connection
  - Get data of all connections
  - Update a connection
- Deal
  - Create a deal
  - Delete a deal
  - Get data of a deal
  - Get data of all deals
  - Update a deal
  - Create a deal note
  - Update a deal note
- E-commerce Order
  - Create a order
  - Delete a order
  - Get data of a order
  - Get data of all orders
  - Update a order
- E-Commerce Customer
  - Create a E-commerce Customer
  - Delete a E-commerce Customer
  - Get data of a E-commerce Customer
  - Get data of all E-commerce Customer
  - Update a E-commerce Customer
- E-commerce Order Products
  - Get data of all order products
  - Get data of a ordered product
  - Get data of an order's products
- List
  - Get all lists
- Tag
  - Create a tag
  - Delete a tag
  - Get data of a tag
  - Get data of all tags
  - Update a tag

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## Adalo node

Use the Adalo node to automate work in Adalo, and integrate Adalo with other applications. n8n has built-in support for a wide range of Adalo features, including like creating, getting, updating and deleting databases, records, and collections.

On this page, you'll find a list of operations the Adalo node supports and links to more resources.

### Operations

- Collection
  - Create
  - Delete
  - Get
  - Get Many
  - Update

## Templates and examples

## Related resources

Refer to [Adalo's documentation](#) for more information on using Adalo. Their [External Collections with APIs](#) page gives more detail about what you can do with Adalo collections.

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## Affinity node

Use the Affinity node to automate work in Affinity, and integrate Affinity with other applications. n8n has built-in support for a wide range of Affinity features, including creating, getting, updating and deleting lists, entries, organization, and persons.

On this page, you'll find a list of operations the Affinity node supports and links to more resources.

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

### Operations

- List
  - Get a list
  - Get all lists
- List Entry
  - Create a list entry
  - Delete a list entry
  - Get a list entry
  - Get all list entries
- Organization
  - Create an organization
  - Delete an organization
  - Get an organization
  - Get all organizations
  - Update an organization
- Person
  - Create a person
  - Delete a person
  - Get a person
  - Get all persons
  - Update a person

## Templates and examples

---

## Agile CRM node

Use the Agile CRM node to automate work in Agile CRM, and integrate Agile CRM with other applications. n8n has built-in support for a wide range of Agile CRM features, including creating, getting, updating and deleting companies, contracts, and deals.

On this page, you'll find a list of operations the Agile CRM node supports and links to more resources.

-8<- “[snippets/integrations/builtin/app-nodes/ai-tools.md](#)”

## Operations

- Company
  - Create a new company
  - Delete a company
  - Get a company
  - Get all companies
  - Update company properties
- Contact
  - Create a new contact
  - Delete a contact
  - Get a contact
  - Get all contacts
  - Update contact properties
- Deal
  - Create a new deal
  - Delete a deal
  - Get a deal
  - Get all deals
  - Update deal properties

## Templates and examples

---

### Airtable node

Use the Airtable node to automate work in Airtable, and integrate Airtable with other applications. n8n has built-in support for a wide range of Airtable features, including creating, reading, listing, updating and deleting tables.

On this page, you'll find a list of operations the Airtable node supports and links to more resources.

### Operations

- Append the data to a table
- Delete data from a table
- List data from a table
- Read data from a table
- Update data in a table

## Templates and examples

### Related resources

n8n provides a trigger node for Airtable. You can find the trigger node docs [here](#).

Refer to [Airtable's documentation](#) for more information about the service.

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

### Node reference

#### Get the Record ID

To fetch data for a particular record, you need the Record ID. There are two ways to get the Record ID.

#### Create a Record ID column in Airtable

To create a Record ID column in your table, refer to this [article](#). You can then use this Record ID in your Airtable node.

#### Use the List operation

To get the Record ID of your record, you can use the **List** operation of the Airtable node. This operation will return the Record ID along with the fields. You can then use this Record ID in your Airtable node.

## Filter records when using the List operation

To filter records from your Airtable base, use the **Filter By Formula** option. For example, if you want to return all the users that belong to the organization n8n, follow the steps mentioned below:

1. Select 'List' from the **Operation** dropdown list.
2. Enter the base ID and the table name in the **Base ID** and **Table** field, respectively.
3. Click on **Add Option** and select 'Filter By Formula' from the dropdown list.
4. Enter the following formula in the **Filter By Formula** field:  
`{Organization}='n8n'`.

Similarly, if you want to return all the users that don't belong to the organization n8n, use the following formula: `NOT({Organization}='n8n')`.

Refer to the Airtable [documentation](#) to learn more about the formulas.

## Common issues

For common errors or issues and suggested resolution steps, refer to [Common Issues](#).

---

## Airtable node common issues

Here are some common errors and issues with the [Airtable node](#) and steps to resolve or troubleshoot them.

### Forbidden - perhaps check your credentials

This error displays when trying to perform actions not permitted by your current level of access. The full text looks something like this:

There was a problem loading the parameter options from server:  
"Forbidden - perhaps check your credentials?"

The error most often displays when the credential you're using doesn't have the scopes it requires on the resources you're attempting to manage.

Refer to the [Airtable credentials](#) and [Airtables scopes](#) documentation for more information.

### Service is receiving too many requests from you

Airtable has a hard API limit on the number of requests generated using personal access tokens.

If you send more than five requests per second per base, you will receive a 429 error, indicating that you have sent too many requests. You will have to wait 30 seconds before resuming requests. This same limit applies for sending more than 50 requests across all bases per access token.

You can find out more in the [Airtable's rate limits documentation](#). If you find yourself running into rate limits with the Airtop node, consider implementing one of the suggestions on the [handling rate limits page](#).

---

## Airtop node

Use the Airtop node to automate work in Airtop, and integrate Airtop with other applications. n8n has built-in support for a wide range of Airtop features, enabling you to control a cloud-based web browser for tasks like querying, scraping, and interacting with web pages.

On this page, you'll find a list of operations the Airtop node supports, and links to more resources.

## Operations

- Session
  - Create session
  - Save profile on termination
  - Terminate session
- Window
  - Create a new browser window
  - Load URL
  - Take screenshot
  - Close window
- Extraction
  - Query page
  - Query page with pagination
  - Smart scrape page
- Interaction
  - Click an element
  - Hover on an element
  - Type

## Templates and examples

## Related resources

Refer to [Airtop's documentation](#) for more information about the service.

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Contact [Airtop's Support](#) for assistance or to create a feature request.

## Node reference

### Create a session and window

Create an Airtop browser session to get a **Session ID**, then use it to create a new browser window. After this, you can use any extraction or interaction operation.

### Extract content

Extract content from a web browser using these operations:

- **Query page**: Extract information from the current window.
- **Query page with pagination**: Extract information from pages with pagination or infinite scrolling.
- **Smart scrape page**: Get the window content as markdown.

Get JSON responses by using the **JSON Output Schema** parameter in query operations.

### Interacting with pages

Click, hover, or type on elements by describing the element you want to interact with.

### Terminate a session

End your session to save resources. Sessions are automatically terminated based on the **Idle Timeout** set in the **Create Session** operation or can be manually terminated using the **Terminate Session** operation.

---

## AMQP Sender node

Use the AMQP Sender node to automate work in AMQP Sender, and integrate AMQP Sender with other applications. n8n has built-in support for a wide range of AMQP Sender features, including sending messages.

On this page, you'll find a list of operations the AMQP Sender node supports and links to more resources.

### Operations

- Send message

### Templates and examples

---

## Anthropic node

Use the Anthropic node to automate work in Anthropic and integrate Anthropic with other applications. n8n has built-in support for a wide range of Anthropic features, including analyzing, uploading, getting, and deleting documents, files, and images, and generating, improving, or templatizing prompts.

On this page, you'll find a list of operations the Anthropic node supports, and links to more resources.

## Operations

- Document:
  - Analyze Document: Take in documents and answer questions about them.
- File:
  - Upload File: Upload a file to the Anthropic API for later user.
  - Get File Metadata: Get metadata for a file from the Anthropic API.
  - List Files: List files from the Anthropic API.
  - Delete File: Delete a file from the Anthropic API.
- Image:
  - Analyze Image: Take in images and answer questions about them.
- Prompt:
  - Generate Prompt: Generate a prompt for a model.
  - Improve Prompt: Improve a prompt for a model.
  - Templatize Prompt: Templatize a prompt for a model.
- Text:
  - Message a Model: Create a completion with an Anthropic model.

## Templates and examples

## Related resources

Refer to [Anthropic's documentation](#) for more information about the service.

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## APITemplate.io node

Use the APITemplate.io node to automate work in APITemplate.io, and integrate APITemplate.io with other applications. n8n has built-in support for a wide range of APITemplate.io features, including getting and creating accounts and PDF.

On this page, you'll find a list of operations the APITemplate.io node supports and links to more resources.

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

## Operations

- Account
  - Get
- Image
  - Create
- PDF
  - Create

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## Asana node

Use the Asana node to automate work in Asana, and integrate Asana with other applications. n8n has built-in support for a wide range of Asana features, including creating, updating, deleting, and getting users, tasks, projects, and subtasks.

On this page, you'll find a list of operations the Asana node supports and links to more resources.

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

## Operations

- Project
  - Create a new project
  - Delete a project
  - Get a project
  - Get all projects
  - Update a project
- Subtask
  - Create a subtask
  - Get all subtasks
- Task
  - Create a task
  - Delete a task
  - Get a task
  - Get all tasks
  - Move a task
  - Search for tasks
  - Update a task
- Task Comment
  - Add a comment to a task
  - Remove a comment from a task
- Task Tag
  - Add a tag to a task
  - Remove a tag from a task
- Task Project
  - Add a task to a project
  - Remove a task from a project
- User

- Get a user
- Get all users

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## Autopilot node

Use the Autopilot node to automate work in Autopilot, and integrate Autopilot with other applications. n8n has built-in support for a wide range of Autopilot features, including creating, deleting, and updating contacts, as well as adding contacts to a list.

On this page, you’ll find a list of operations the Autopilot node supports and links to more resources.

## Operations

- Contact
  - Create/Update a contact
  - Delete a contact
  - Get a contact
  - Get all contacts
- Contact Journey
  - Add contact to list
- Contact List
  - Add contact to list
  - Check if contact is on list
  - Get all contacts on list
  - Remove a contact from a list
- List
  - Create a list
  - Get all lists

## Templates and examples

## AWS Certificate Manager node

Use the AWS Certificate Manager node to automate work in AWS Certificate Manager, and integrate AWS Certificate Manager with other applications. n8n has built-in support for a wide range of AWS Certificate Manager features, including creating, deleting, getting, and renewing SSL certificates.

On this page, you’ll find a list of operations the AWS Certificate Manager node supports and links to more resources.

## Operations

- Certificate
  - Delete
  - Get
  - Get Many
  - Get Metadata
  - Renew

## Templates and examples

## Related resources

Refer to [AWS Certificate Manager's documentation](#) for more information on this service.

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## AWS Cognito node

Use the AWS Cognito node to automate work in AWS Cognito and integrate AWS Cognito with other applications. n8n has built-in support for a wide range of AWS Cognito features, which includes creating, retrieving, updating, and deleting groups, users, and user pools.

On this page, you'll find a list of operations the AWS Cognito node supports, and links to more resources.

## Operations

- Group:
  - Create: Create a new group.
  - Delete: Delete an existing group.
  - Get: Retrieve details about an existing group.
  - Get Many: Retrieve a list of groups.
  - Update: Update an existing group.
- User:
  - Add to Group: Add an existing user to a group.
  - Create: Create a new user.
  - Delete: Delete a user.
  - Get: Retrieve information about an existing user.
  - Get Many: Retrieve a list of users.
  - Remove From Group: Remove a user from a group.
  - Update: Update an existing user.
- User Pool:
  - Get: Retrieve information about an existing user pool.

## Templates and examples

## Related resources

Refer to [AWS Cognito's documentation](#) for more information about the service.

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## AWS Comprehend node

Use the AWS Comprehend node to automate work in AWS Comprehend, and integrate AWS Comprehend with other applications. n8n has built-in support for a wide range of AWS Comprehend features, including identifying and analyzing texts.

On this page, you'll find a list of operations the AWS Comprehend node supports and links to more resources.

### Operations

#### Text

- Identify the dominant language
- Analyse the sentiment of the text

### Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## AWS DynamoDB node

Use the AWS DynamoDB node to automate work in AWS DynamoDB, and integrate AWS DynamoDB with other applications. n8n has built-in support for a wide range of AWS DynamoDB features, including creating, reading, updating, deleting items, and records on a database.

On this page, you'll find a list of operations the AWS DynamoDB node supports and links to more resources.

### Operations

- Item
  - Create a new record, or update the current one if it already exists (upsert/put)
  - Delete an item
  - Get an item
  - Get all items

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## AWS Elastic Load Balancing node

Use the AWS Elastic Load Balancing node to automate work in AWS ELB, and integrate AWS ELB with other applications. n8n has built-in support for a wide range of AWS ELB features, including adding, getting, removing, deleting certificates and load balancers.

On this page, you’ll find a list of operations the AWS ELB node supports and links to more resources.

## Operations

- Listener Certificate
  - Add
  - Get Many
  - Remove
- Load Balancer
  - Create
  - Delete
  - Get
  - Get Many

This node supports creating and managing application and network load balancers. It doesn’t currently support gateway load balancers.

## Templates and examples

## Related resources

Refer to [AWS ELB’s documentation](#) for more information on this service.

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## AWS IAM node

Use the AWS IAM node to automate work in AWS Identity and Access Management (IAM) and integrate AWS IAM with other applications. n8n has built-in support for a wide range of AWS IAM features, which includes creating, updating, getting and deleting users and groups as well as managing group membership.

On this page, you’ll find a list of operations the AWS IAM node supports, and links to more resources.

## Operations

- **User:**
  - **Add to Group:** Add an existing user to a group.
  - **Create:** Create a new user.
  - **Delete:** Delete a user.
  - **Get:** Retrieve a user.
  - **Get Many:** Retrieve a list of users.
  - **Remove From Group:** Remove a user from a group.
  - **Update:** Update an existing user.
- **Group:**
  - **Create:** Create a new group.
  - **Delete:** Create a new group.
  - **Get:** Retrieve a group.
  - **Get Many:** Retrieve a list of groups.
  - **Update:** Update an existing group.

## Templates and examples

## Related resources

Refer to the [AWS IAM documentation](#) for more information about the service.

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## AWS Lambda node

Use the AWS Lambda node to automate work in AWS Lambda, and integrate AWS Lambda with other applications. n8n has built-in support for a wide range of AWS Lambda features, including invoking functions.

On this page, you'll find a list of operations the AWS Lambda node supports and links to more resources.

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

## Operations

- Invoke a function

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

# AWS Rekognition node

Use the AWS Rekognition node to automate work in AWS Rekognition, and integrate AWS Rekognition with other applications. n8n has built-in support for a wide range of AWS Rekognition features, including analyzing images.

On this page, you'll find a list of operations the AWS Rekognition node supports and links to more resources.

## Operations

### Image

- Analyze

## Templates and examples

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

---

# AWS S3 node

Use the AWS S3 node to automate work in AWS S3, and integrate AWS S3 with other applications. n8n has built-in support for a wide range of AWS S3 features, including creating and deleting buckets, copying and downloading files, as well as getting folders.

On this page, you'll find a list of operations the AWS S3 node supports and links to more resources.

```
-8<- "_snippets/integrations/builtin/app-nodes/ai-tools.md"
```

## Operations

- Bucket
  - Create a bucket
  - Delete a bucket
  - Get all buckets
  - Search within a bucket
- File
  - Copy a file
  - Delete a file
  - Download a file
  - Get all files
  - Upload a file
- Folder
  - Create a folder
  - Delete a folder
  - Get all folders

## Templates and examples

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

---

## AWS SES node

Use the AWS SES node to automate work in AWS SES, and integrate AWS SES with other applications. n8n has built-in support for a wide range of AWS SES features, including creating, getting, deleting, sending, updating, and adding templates and emails.

On this page, you'll find a list of operations the AWS SES node supports and links to more resources.

```
-8<- "_snippets/integrations/builtin/app-nodes/ai-tools.md"
```

### Operations

- Custom Verification Email
  - Create a new custom verification email template
  - Delete an existing custom verification email template
  - Get the custom email verification template
  - Get all the existing custom verification email templates for your account
  - Add an email address to the list of identities
  - Update an existing custom verification email template.
- Email
  - Send
  - Send Template
- Template
  - Create a template
  - Delete a template
  - Get a template
  - Get all templates
  - Update a template

### Templates and examples

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

---

## AWS SNS node

Use the AWS SNS node to automate work in AWS SNS, and integrate AWS SNS with other applications. n8n has built-in support for a wide range of AWS SNS features, including publishing messages.

On this page, you'll find a list of operations the AWS SNS node supports and links to more resources.

### Operations

- Publish a message to a topic

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## AWS SQS node

Use the AWS SQS node to automate work in AWS SNS, and integrate AWS SQS with other applications. n8n has built-in support for a wide range of AWS SQS features, including sending messages.

On this page, you'll find a list of operations the AWS SQS node supports and links to more resources.

### Operations

- Send a message to a queue.

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## AWS Textract node

Use the AWS Textract node to automate work in AWS Textract, and integrate AWS Textract with other applications. n8n has built-in support for a wide range of AWS Textract features, including analyzing invoices.

On this page, you'll find a list of operations the AWS Textract node supports and links to more resources.

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

### Operations

- Analyze Receipt or Invoice

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

# AWS Transcribe node

Use the AWS Transcribe node to automate work in AWS Transcribe, and integrate AWS Transcribe with other applications. n8n has built-in support for a wide range of AWS Transcribe features, including creating, deleting, and getting transcription jobs.

On this page, you'll find a list of operations the AWS Transcribe node supports and links to more resources.

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

## Operations

### Transcription Job

- Create a transcription job
- Delete a transcription job
- Get a transcription job
- Get all transcriptions job

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

# Azure Cosmos DB node

Use the Azure Cosmos DB node to automate work in Azure Cosmos DB and integrate Azure Cosmos DB with other applications. n8n has built-in support for a wide range of Azure Cosmos DB features, which includes creating, getting, updating, and deleting containers and items.

On this page, you'll find a list of operations the Azure Cosmos DB node supports, and links to more resources.

## Operations

- **Container:**
  - **Create**
  - **Delete**
  - **Get**
  - **Get Many**
- **Item:**
  - **Create**
  - **Delete**
  - **Get**
  - **Get Many**
  - **Execute Query**
  - **Update**

## Templates and examples

### Related resources

Refer to [Azure Cosmos DB's documentation](#) for more information about the service.

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

---

## Azure Storage node

The Azure Storage node has built-in support for a wide range of features, which includes creating, getting, and deleting blobs and containers. Use this node to automate work within the Azure Storage service or integrate it with other services in your workflow.

On this page, you'll find a list of operations the Azure Storage node supports, and links to more resources.

### Operations

- **Blob**
  - **Create blob:** Create a new blob or replace an existing one.
  - **Delete blob:** Delete an existing blob.
  - **Get blob:** Retrieve data for a specific blob.
  - **Get many blobs:** Retrieve a list of blobs.
- **Container**
  - **Create container:** Create a new container.
  - **Delete container:** Delete an existing container.
  - **Get container:** Retrieve data for a specific container.
  - **Get many containers:** Retrieve a list of containers.

## Templates and examples

### Related resources

Refer to [Microsoft's Azure Storage documentation](#) for more information about the service.

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

---

## BambooHR node

Use the BambooHR node to automate work in BambooHR, and integrate BambooHR with other applications. n8n has built-in support for a wide range of BambooHR features, including creating, deleting,

downloading, and getting company reports, employee documents, and files.

On this page, you'll find a list of operations the BambooHR node supports and links to more resources.

## Operations

- Company Report
  - Get a company report
- Employee
  - Create an employee
  - Get an employee
  - Get all employees
  - Update an employee
- Employee Document
  - Delete an employee document
  - Download an employee document
  - Get all employee document
  - Update an employee document
  - Upload an employee document
- File
  - Delete a company file
  - Download a company file
  - Get all company files
  - Update a company file
  - Upload a company file

## Templates and examples

---

## Bannerbear node

Use the Bannerbear node to automate work in Bannerbear, and integrate Bannerbear with other applications. n8n has built-in support for a wide range of Bannerbear features, including creating and getting images and templates.

On this page, you'll find a list of operations the Bannerbear node supports and links to more resources.

## Operations

- Image
  - Create an image
  - Get an image
- Template
  - Get a template
  - Get all templates

## Templates and examples

---

# Baserow node

Use the Baserow node to automate work in Baserow, and integrate Baserow with other applications. n8n has built-in support for a wide range of Baserow features, including creating, getting, retrieving, and updating rows.

On this page, you'll find a list of operations the Baserow node supports and links to more resources.

```
-8<- "_snippets/integrations/builtin/app-nodes/ai-tools.md"
```

## Operations

- Row
  - Create a row
  - Delete a row
  - Retrieve a row
  - Retrieve all rows
  - Update a row

## Templates and examples

---

# Beeminder node

Use the Beeminder node to automate work in Beeminder, and integrate Beeminder with other applications. n8n has built-in support for a wide range of Beeminder features, including creating, deleting, and updating data points.

On this page, you'll find a list of operations the Beeminder node supports and links to more resources.

## Operations

**data point** - Create data point for a goal - Delete a data point - Get all data points for a goal - Update a data point

## Templates and examples

---

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

---

# Bitly node

Use the Bitly node to automate work in Bitly, and integrate Bitly with other applications. n8n has built-in support for a wide range of Bitly features, including creating, getting, and updating links.

On this page, you'll find a list of operations the Bitly node supports and links to more resources.

## Operations

- Link
  - Create a link
  - Get a link
  - Update a link

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## Bitwarden node

Use the Bitwarden node to automate work in Bitwarden, and integrate Bitwarden with other applications. n8n has built-in support for a wide range of Bitwarden features, including creating, getting, deleting, and updating collections, events, groups, and members.

On this page, you'll find a list of operations the Bitwarden node supports and links to more resources.

## Operations

- Collection
  - Delete
  - Get
  - Get All
  - Update
- Event
  - Get All
- Group
  - Create
  - Delete
  - Get
  - Get All
  - Get Members
  - Update
  - Update Members
- Member
  - Create
  - Delete
  - Get
  - Get All
  - Get Groups
  - Update
  - Update Groups

## Templates and examples

---

## Box node

Use the Box node to automate work in Box, and integrate Box with other applications. n8n has built-in support for a wide range of Box features, including creating, copying, deleting, searching, uploading, and downloading files and folders.

On this page, you'll find a list of operations the Box node supports and links to more resources.

### Operations

- File
  - Copy a file
  - Delete a file
  - Download a file
  - Get a file
  - Search files
  - Share a file
  - Upload a file
- Folder
  - Create a folder
  - Get a folder
  - Delete a folder
  - Search files
  - Share a folder
  - Update folder

### Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## Brandfetch node

Use the Brandfetch node to automate work in Brandfetch, and integrate Brandfetch with other applications. n8n has built-in support for a wide range of Brandfetch features, including returning a company's information.

On this page, you'll find a list of operations the Brandfetch node supports and links to more resources.

### Operations

- Return a company's colors
- Return a company's data
- Return a company's fonts
- Return a company's industry
- Return a company's logo & icon

## Templates and examples

---

### Brevo node

Use the Brevo node to automate work in Brevo, and integrate Brevo with other applications. n8n has built-in support for a wide range of Brevo features, including creating, updating, deleting, and getting contacts, attributes, as well as sending emails.

On this page, you'll find a list of operations the Brevo node supports and links to more resources.

### Operations

- Contact
  - Create
  - Create or Update
  - Delete
  - Get
  - Get All
  - Update
- Contact Attribute
  - Create
  - Delete
  - Get All
  - Update
- Email
  - Send
  - Send Template
- Sender
  - Create
  - Delete
  - Get All

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

### Bubble node

Use the Bubble node to automate work in Bubble, and integrate Bubble with other applications. n8n has built-in support for a wide range of Bubble features, including creating, deleting, getting, and updating objects.

On this page, you'll find a list of operations the Bubble node supports and links to more resources.

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

## Operations

- Object
  - Create
  - Delete
  - Get
  - Get All
  - Update

## Templates and examples

---

## Chargebee node

Use the Chargebee node to automate work in Chargebee, and integrate Chargebee with other applications. n8n has built-in support for a wide range of Chargebee features, including creating customers, returning invoices, and canceling subscriptions.

On this page, you'll find a list of operations the Chargebee node supports and links to more resources.

## Operations

- Customer
  - Create a customer
- Invoice
  - Return the invoices
  - Get URL for the invoice PDF
- Subscription
  - Cancel a subscription
  - Delete a subscription

## Templates and examples

---

## CircleCI node

Use the CircleCI node to automate work in CircleCI, and integrate CircleCI with other applications. n8n has built-in support for a wide range of CircleCI features, including getting and triggering pipelines.

On this page, you'll find a list of operations the CircleCI node supports and links to more resources.

## Operations

- Pipeline
  - Get a pipeline
  - Get all pipelines
  - Trigger a pipeline

## Templates and examples

---

### Webex by Cisco node

Use the Webex by Cisco node to automate work in Webex, and integrate Webex with other applications. n8n has built-in support for a wide range of Webex features, including creating, getting, updating, and deleting meetings and messages.

On this page, you'll find a list of operations the Webex node supports and links to more resources.

### Operations

- Meeting
  - Create
  - Delete
  - Get
  - Get All
  - Update
- Message
  - Create
  - Delete
  - Get
  - Get All
  - Update

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

### Clearbit node

Use the Clearbit node to automate work in Clearbit, and integrate Clearbit with other applications. n8n has built-in support for a wide range of Clearbit features, including autocompleting and looking up companies and persons.

On this page, you'll find a list of operations the Clearbit node supports and links to more resources.

### Operations

- Company
  - Auto-complete company names and retrieve logo and domain
  - Look up person and company data based on an email or domain
- Person
  - Look up a person and company data based on an email or domain

## Templates and examples

---

### ClickUp node

Use the ClickUp node to automate work in ClickUp, and integrate ClickUp with other applications. n8n has built-in support for a wide range of ClickUp features, including creating, getting, deleting, and updating folders, checklists, tags, comments, and goals.

On this page, you'll find a list of operations the ClickUp node supports and links to more resources.

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

### Operations

- Checklist
  - Create a checklist
  - Delete a checklist
  - Update a checklist
- Checklist Item
  - Create a checklist item
  - Delete a checklist item
  - Update a checklist item
- Comment
  - Create a comment
  - Delete a comment
  - Get all comments
  - Update a comment
- Folder
  - Create a folder
  - Delete a folder
  - Get a folder
  - Get all folders
  - Update a folder
- Goal
  - Create a goal
  - Delete a goal
  - Get a goal
  - Get all goals
  - Update a goal
- Goal Key Result
  - Create a key result
  - Delete a key result
  - Update a key result
- List
  - Create a list
  - Retrieve list's custom fields
  - Delete a list
  - Get a list
  - Get all lists
  - Get list members
  - Update a list
- Space Tag
  - Create a space tag
  - Delete a space tag

- Get all space tags
- Update a space tag
- Task
  - Create a task
  - Delete a task
  - Get a task
  - Get all tasks
  - Get task members
  - Set a custom field
  - Update a task
- Task List
  - Add a task to a list
  - Remove a task from a list
- Task Tag
  - Add a tag to a task
  - Remove a tag from a task
- Task Dependency
  - Create a task dependency
  - Delete a task dependency
- Time Entry
  - Create a time entry
  - Delete a time entry
  - Get a time entry
  - Get all time entries
  - Start a time entry
  - Stop the current running timer
  - Update a time Entry
- Time Entry Tag
  - Add tag to time entry
  - Get all time entry tags
  - Remove tag from time entry

## Operation details

### Get a task

When using the **Get a task** operation, you can optionally enable the following:

- **Include Subtasks:** When enabled, also fetches and includes subtasks for the specified task.
- **Include Markdown Description:** When enabled, includes the `markdown_description` field in the response, which preserves links and formatting in the task description. This is useful if your task descriptions contain links or rich formatting.

### Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## Clockify node

Use the Clockify node to automate work in Clockify, and integrate Clockify with other applications. n8n has built-in support for a wide range of Clockify features, including creating, updating, getting, and deleting tasks, time entries, projects, and tags.

On this page, you'll find a list of operations the Clockify node supports and links to more resources.

## Operations

- Project
  - Create a project
  - Delete a project
  - Get a project
  - Get all projects
  - Update a project
- Tag
  - Create a tag
  - Delete a tag
  - Get all tags
  - Update a tag
- Task
  - Create a task
  - Delete a task
  - Get a task
  - Get all tasks
  - Update a task
- Time Entry
  - Create a time entry
  - Delete a time entry
  - Get time entry
  - Update a time entry

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## Cloudflare node

Use the Cloudflare node to automate work in Cloudflare, and integrate Cloudflare with other applications. n8n has built-in support for a wide range of Cloudflare features, including deleting, getting, and uploading zone certificates.

On this page, you'll find a list of operations the Cloudflare node supports and links to more resources.

## Operations

- Zone Certificate
  - Delete
  - Get
  - Get Many

- Upload

## Templates and examples

### Related resources

Refer to [Cloudflare's API documentation on zone-level authentication](#) for more information on this service.

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## Cockpit node

Use the Cockpit node to automate work in Cockpit, and integrate Cockpit with other applications. n8n has built-in support for a wide range of Cockpit features, including creating a collection entry, storing data from a form submission, and getting singletons.

On this page, you'll find a list of operations the Cockpit node supports and links to more resources.

### Operations

- Collection
  - Create a collection entry
  - Get all collection entries
  - Update a collection entry
- Form
  - Store data from a form submission
- Singleton
  - Get a singleton

## Templates and examples

---

## Coda node

Use the Coda node to automate work in Coda, and integrate Coda with other applications. n8n has built-in support for a wide range of Coda features, including creating, getting, and deleting controls, formulas, tables, and views.

On this page, you'll find a list of operations the Coda node supports and links to more resources.

### Operations

- Control

- Get a control
  - Get all controls
- Formula
  - Get a formula
  - Get all formulas
- Table
  - Create/Insert a row
  - Delete one or multiple rows
  - Get all columns
  - Get all the rows
  - Get a column
  - Get a row
  - Pushes a button
- View
  - Delete view row
  - Get a view
  - Get all views
  - Get all views columns
  - Get all views rows
  - Update row
  - Push view button

## Templates and examples

---

### CoinGecko node

Use the CoinGecko node to automate work in CoinGecko, and integrate CoinGecko with other applications. n8n has built-in support for a wide range of CoinGecko features, including getting coins and events.

On this page, you'll find a list of operations the CoinGecko node supports and links to more resources.

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

### Operations

- Coin
  - Get a candlestick open-high-low-close chart for the selected currency
  - Get current data for a coin
  - Get all coins
  - Get historical data (name, price, market, stats) at a given date for a coin
  - Get prices and market related data for all trading pairs that match the selected currency
  - Get historical market data include price, market cap, and 24h volume (granularity auto)
  - Get the current price of any cryptocurrencies in any other supported currencies that you need
  - Get coin tickers
- Event
  - Get all events

## Templates and examples

---

## Contentful node

Use the Contentful node to automate work in Contentful, and integrate Contentful with other applications. n8n has built-in support for a wide range of Contentful features, including getting assets, content types, entries, locales, and space.

On this page, you'll find a list of operations the Contentful node supports and links to more resources.

### Operations

- Asset
  - Get
  - Get All
- Content Type
  - Get
- Entry
  - Get
  - Get All
- Locale
  - Get All
- Space
  - Get

## Templates and examples

---

## ConvertKit node

Use the ConvertKit node to automate work in ConvertKit, and integrate ConvertKit with other applications. n8n has built-in support for a wide range of ConvertKit features, including creating and deleting custom fields, getting tags, and adding subscribers.

On this page, you'll find a list of operations the ConvertKit node supports and links to more resources.

### Operations

- Custom Field
  - Create a field
  - Delete a field
  - Get all fields
  - Update a field
- Form
  - Add a subscriber
  - Get all forms
  - List subscriptions to a form including subscriber data

- Sequence
  - Add a subscriber
  - Get all sequences
  - Get all subscriptions to a sequence including subscriber data
- Tag
  - Create a tag
  - Get all tags
- Tag Subscriber
  - Add a tag to a subscriber
  - List subscriptions to a tag including subscriber data
  - Delete a tag from a subscriber

## Templates and examples

---

## Copper node

Use the Copper node to automate work in Copper, and integrate Copper with other applications. n8n has built-in support for a wide range of Copper features, including getting, updating, deleting, and creating companies, customer sources, leads, projects and tasks.

On this page, you'll find a list of operations the Copper node supports and links to more resources.

## Operations

- Company
  - Create
  - Delete
  - Get
  - Get All
  - Update
- Customer Source
  - Get All
- Lead
  - Create
  - Delete
  - Get
  - Get All
  - Update
- Opportunity
  - Create
  - Delete
  - Get
  - Get All
  - Update
- Person
  - Create
  - Delete
  - Get
  - Get All
  - Update
- Project
  - Create
  - Delete

- Get
- Get All
- Update
- Task
  - Create
  - Delete
  - Get
  - Get All
  - Update
- User
  - Get All

## Templates and examples

---

## Cortex node

Use the Cortex node to automate work in Cortex, and integrate Cortex with other applications. n8n has built-in support for a wide range of Cortex features, including executing analyzers, and responders, as well as getting job details.

On this page, you'll find a list of operations the Cortex node supports and links to more resources.

## Operations

- Analyzer
  - Execute Analyzer
- Job
  - Get job details
  - Get job report
- Responder
  - Execute Responder

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## CrateDB node

Use the CrateDB node to automate work in CrateDB, and integrate CrateDB with other applications. n8n has built-in support for a wide range of CrateDB features, including executing, inserting, and updating rows in the database.

On this page, you'll find a list of operations the CrateDB node supports and links to more resources.

## Operations

- Execute an SQL query
- Insert rows in database
- Update rows in database

## Templates and examples

### Node reference

#### Specify a column's data type

To specify a column's data type, append the column name with :type, where type is the data type you want for the column. For example, if you want to specify the type int for the column **id** and type text for the column **name**, you can use the following snippet in the **Columns** field: id:int,name:text.

---

## crowd.dev node

Use the crowd.dev node to automate work in crowd.dev and integrate crowd.dev with other applications. n8n has built-in support for a wide range of crowd.dev features, which includes creating, updating, and deleting members, notes, organizations, and tasks.

On this page, you'll find a list of operations the crowd.dev node supports, and links to more resources.

## Operations

- Activity
  - Create or Update with a Member
  - Create
- Automation
  - Create
  - Destroy
  - Find
  - List
  - Update
- Member
  - Create or Update
  - Delete
  - Find
  - Update
- Note
  - Create
  - Delete
  - Find
  - Update
- Organization
  - Create
  - Delete

- Find
- Update
- Task
  - Create
  - Delete
  - Find
  - Update

## Templates and examples

### Related resources

n8n provides a trigger node for crowd.dev. You can find the trigger node docs [here](#).

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## Customer.io node

Use the Customer.io node to automate work in Customer.io, and integrate Customer.io with other applications. n8n has built-in support for a wide range of Customer.io features, including creating and updating customers, tracking events, and getting campaigns.

On this page, you'll find a list of operations the Customer.io node supports and links to more resources.

### Operations

- Customer
  - Create/Update a customer.
  - Delete a customer.
- Event
  - Track a customer event.
  - Track an anonymous event.
- Campaign
  - Get
  - Get All
  - Get Metrics
- Segment
  - Add Customer
  - Remove Customer

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## DeepL node

Use the DeepL node to automate work in DeepL, and integrate DeepL with other applications. n8n has built-in support for a wide range of DeepL features, including translating languages.

On this page, you'll find a list of operations the DeepL node supports and links to more resources.

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

## Operations

- Language
  - Translate data

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## Demio node

Use the Demio node to automate work in Demio, and integrate Demio with other applications. n8n has built-in support for a wide range of Demio features, including getting, and registering events and reports.

On this page, you'll find a list of operations the Demio node supports and links to more resources.

## Operations

- Event
  - Get an event
  - Get all events
  - Register someone to an event
- Report
  - Get an event report

## Templates and examples

## DHL node

Use the DHL node to automate work in DHL, and integrate DHL with other applications. n8n has built-in support for a wide range of DHL features, including tracking shipment.

On this page, you'll find a list of operations the DHL node supports and links to more resources.

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

## Operations

- Shipment
  - Get Tracking Details

## Templates and examples

---

## Discord node

Use the Discord node to automate work in Discord, and integrate Discord with other applications. n8n has built-in support for a wide range of Discord features, including sending messages in a Discord channel and managing channels.

On this page, you'll find a list of operations the Discord node supports and links to more resources.

-8<- “[\\_ snippets/integrations/builtin/app-nodes/ai-tools.md](#)”

## Operations

- Channel
  - Create
  - Delete
  - Get
  - Get Many
  - Update
- Message
  - Delete
  - Get
  - Get Many
  - React with Emoji
  - Send
  - Send and Wait for Response
- Member
  - Get Many
  - Role Add
  - Role Remove

-8<- “[\\_ snippets/integrations/builtin/send-and-wait-operation.md](#)”

## Templates and examples

## Related resources

Refer to [Discord's documentation](#) for more information about the service.

-8<- “[\\_ snippets/integrations/builtin/app-nodes/operation-not-supported.md](#)”

## Common issues

For common errors or issues and suggested resolution steps, refer to [Common Issues](#).

---

## Discord node common issues

Here are some common errors and issues with the [Discord node](#) and steps to resolve or troubleshoot them.

### Add extra fields to embeds

Discord messages can optionally include embeds, a rich preview component that can include a title, description, image, link, and more.

The Discord node supports embeds when using the **Send** operation on the **Message** resource. Select **Add Embeds** to set extra fields including Description, Author, Title, URL, and URL Image.

To add fields that aren't included by default, set **Input Method** to **Raw JSON**. From here, add a JSON object to the **Value** parameter defining the field names and values you want to include.

For example, to include footer and fields, neither of which are available using the **Enter Fields** Input Method, you could use a JSON object like this:

```
{
 "author": "My Name",
 "url": "https://discord.js.org",
 "fields": [
 {
 "name": "Regular field title",
 "value": "Some value here"
 }
],
 "footer": {
 "text": "Some footer text here",
 "icon_url": "https://i.imgur.com/AfFp7pu.png"
 }
}
```

You can learn more about embeds in [Using Webhooks and Embeds | Discord](#).

If you experience issues when working with embeds with the Discord node, you can use the [HTTP Request](#) with your existing Discord credentials to POST to the following URL:

[https://discord.com/api/v10/channels/<CHANNEL\\_ID>/messages](https://discord.com/api/v10/channels/<CHANNEL_ID>/messages)

In the body, include your embed information in the message content like this:

```
{
 "content": "Test",
 "embeds": [
```

```

{
 "author": "My Name",
 "url": "https://discord.js.org",
 "fields": [
 {
 "name": "Regular field title",
 "value": "Some value here"
 }
],
 "footer": {
 "text": "Some footer text here",
 "icon_url": "https://i.imgur.com/AfFp7pu.png"
 }
}
]
}

```

## Mention users and channels

To mention users and channels in Discord messages, you need to format your message according to [Discord's message formatting guidelines](#).

To mention a user, you need to know the Discord user's user ID. Keep in mind that the user ID is different from the user's display name. Similarly, you need a channel ID to link to a specific channel.

You can learn how to enable developer mode and copy the user or channel IDs in [Discord's documentation on finding User/Server/Message IDs](#).

Once you have the user or channel ID, you can format your message with the following syntax:

- **User:** <@USER\_ID>
  - **Channel:** <#CHANNEL\_ID>
  - **Role:** <@&ROLE\_ID>
- 

## Discourse node

Use the Discourse node to automate work in Discourse, and integrate Discourse with other applications. n8n has built-in support for a wide range of Discourse features, including creating, getting, updating, and removing categories, groups, posts, and users.

On this page, you'll find a list of operations the Discourse node supports and links to more resources.

## Operations

- Category
  - Create a category
  - Get all categories
  - Update a category
- Group

- Create a group
- Get a group
- Get all groups
- Update a group
- Post
  - Create a post
  - Get a post
  - Get all posts
  - Update a post
- User
  - Create a user
  - Get a user
  - Get all users
- User Group
  - Create a user to group
  - Remove user from group

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## Disqus node

Use the Disqus node to automate work in Disqus, and integrate Disqus with other applications. n8n has built-in support for a wide range of Disqus features, including returning forums.

On this page, you’ll find a list of operations the Disqus node supports and links to more resources.

## Operations

- Forum
  - Return forum details
  - Return a list of categories within a forum
  - Return a list of threads within a forum
  - Return a list of posts within a forum

## Templates and examples

---

## Drift node

Use the Drift node to automate work in Drift, and integrate Drift with other applications. n8n has built-in support for a wide range of Drift features, including creating, updating, deleting, and getting contacts.

On this page, you’ll find a list of operations the Drift node supports and links to more resources.

## Operations

- Contact
  - Create a contact
  - Get custom attributes
  - Delete a contact
  - Get a contact
  - Update a contact

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## Dropbox node

Use the Dropbox node to automate work in Dropbox, and integrate Dropbox with other applications. n8n has built-in support for a wide range of Dropbox features, including creating, downloading, moving, and copying files and folders.

On this page, you'll find a list of operations the Dropbox node supports and links to more resources.

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

## Operations

- File
  - Copy a file
  - Delete a file
  - Download a file
  - Move a file
  - Upload a file
- Folder
  - Copy a folder
  - Create a folder
  - Delete a folder
  - Return the files and folders in a given folder
  - Move a folder
- Search
  - Query

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## Dropcontact node

Use the Dropcontact node to automate work in Dropcontact, and integrate Dropcontact with other applications. n8n has built-in support for a wide range of Dropcontact features, including fetching contacts.

On this page, you'll find a list of operations the Dropcontact node supports and links to more resources.

## Operations

**Contact** - Enrich - Fetch Request

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## E-goi node

Use the E-goi node to automate work in E-goi, and integrate E-goi with other applications. n8n has built-in support for a wide range of E-goi features, including creating, updating, deleting, and getting contacts.

On this page, you'll find a list of operations the E-goi node supports and links to more resources.

## Operations

**Contact**

- Create a member
- Get a member
- Get all members
- Update a member

## Templates and examples

## Elasticsearch node

Use the Elasticsearch node to automate work in Elasticsearch, and integrate Elasticsearch with other applications. n8n has built-in support for a wide range of Elasticsearch features, including creating, updating, deleting, and getting documents and indexes.

On this page, you'll find a list of operations the Elasticsearch node supports and links to more resources.

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

## Operations

- Document
  - Create a document
  - Delete a document
  - Get a document
  - Get all documents
  - Update a document
- Index
  - Create
  - Delete
  - Get
  - Get All

## Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## Elastic Security node

Use the Elastic Security node to automate work in Elastic Security, and integrate Elastic Security with other applications. n8n's has built-in support for a wide range of Elastic Security features, including creating, updating, deleting, retrieving, and getting cases.

On this page, you'll find a list of operations the Elastic Security node supports and links to more resources.

## Operations

- Case
  - Create a case
  - Delete a case
  - Get a case
  - Retrieve all cases
  - Retrieve a summary of all case activity
  - Update a case
- Case Comment
  - Add a comment to a case
  - Get a case comment
  - Retrieve all case comments
  - Remove a comment from a case
  - Update a comment in a case
- Case Tag
  - Add a tag to a case
  - Remove a tag from a case
- Connector
  - Create a connector

## Templates and examples

---

## Emelia node

Use the Emelia node to automate work in Emelia, and integrate Emelia with other applications. n8n has built-in support for a wide range of Emelia features, including creating campaigns, and adding contacts to a list.

On this page, you'll find a list of operations the Emelia node supports and links to more resources.

### Operations

- Campaign
  - Add Contact
  - Create
  - Get
  - Get All
  - Pause
  - Start
- Contact List
  - Add
  - Get All

### Templates and examples

---

## ERPNext node

Use the ERPNext node to automate work in ERPNext, and integrate ERPNext with other applications. n8n has built-in support for a wide range of ERPNext features, including creating, updating, retrieving, and deleting documents.

On this page, you'll find a list of operations the ERPNext node supports and links to more resources.

-8<- “\_snippets/integrations/builtin/app-nodes/ai-tools.md”

### Operations

**Document** - Create a document - Delete a document - Retrieve a document - Retrieve all documents - Update a document

### Templates and examples

-8<- “\_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

---

## Facebook Graph API node

Use the Facebook Graph API node to automate work in Facebook Graph API, and integrate Facebook Graph API with other applications. n8n has built-in support for a wide range of Facebook Graph API features, including using queries GET POST DELETE for several parameters like host URL, request methods and much more.

On this page, you'll find a list of operations the Facebook Graph API node supports and links to more resources.

## Operations

- **Default**
  - GET
  - POST
  - DELETE
- **Video Uploads**
  - GET
  - POST
  - DELETE

## Parameters

- ```
-8<- "_snippets/integrations/builtin/app-nodes/ai-tools.md"
```
- **Host URL:** The host URL for the request. The following options are available:
 - **Default:** Requests are passed to the graph.facebook.com host URL. Used for the majority of requests.
 - **Video:** Requests are passed to the graph-video.facebook.com host URL. Used for video upload requests only.
 - **HTTP Request Method:** The method to be used for this request, from the following options:
 - GET
 - POST
 - DELETE
 - **Graph API Version:** The version of the Facebook Graph API to be used for this request.
 - **Node:** The node on which to operate, for example /<page-id>/feed. Read more about it in the official Facebook Developer documentation.
 - **Edge:** Edge of the node on which to operate. Edges represent collections of objects which are attached to the node.
 - **Ignore SSL Issues:** Toggle to still download the response even if SSL certificate validation isn't possible.
 - **Send Binary File:** Available for POST operations. If enabled binary data is sent as the body. Requires setting the following:
 - **Input Binary Field:** Name of the binary property which contains the data for the file to be uploaded.

Templates and examples

FileMaker node

Use the FileMaker node to automate work in FileMaker, and integrate FileMaker with other applications. n8n has built-in support for a wide range of FileMaker features, including creating, finding, getting, editing, and duplicating files.

On this page, you'll find a list of operations the FileMaker node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Find Records
- Get Records
- Get Records by Id
- Perform Script
- Create Record
- Edit Record
- Duplicate Record
- Delete Record

Templates and examples

Flow node

Use the Flow node to automate work in Flow, and integrate Flow with other applications. n8n has built-in support for a wide range of Flow features, including creating, updating, and getting tasks.

On this page, you'll find a list of operations the Flow node supports and links to more resources.

Operations

- Task
 - Create a new task
 - Update a task
 - Get a task
 - Get all the tasks

Templates and examples

Freshdesk node

Use the Freshdesk node to automate work in Freshdesk and integrate Freshdesk with other applications. n8n has built-in support for a wide range of Freshdesk features, including creating, updating, deleting, and getting contacts and tickets.

On this page, you'll find a list of operations the Freshdesk node supports and links to more resources.

Operations

- Contact
 - Create a new contact
 - Delete a contact
 - Get a contact
 - Get all contacts
 - Update a contact
- Ticket
 - Create a new ticket
 - Delete a ticket
 - Get a ticket
 - Get all tickets
 - Update a ticket

Templates and examples

Freshservice node

Use the Freshservice node to automate work in Freshservice and integrate Freshservice with other applications. n8n has built-in support for a wide range of Freshdesk features, including creating, updating, deleting, and getting agent information and departments.

On this page, you'll find a list of operations the Freshservice node supports and links to more resources.

Operations

- Agent
 - Create an agent
 - Delete an agent
 - Retrieve an agent
 - Retrieve all agents
 - Update an agent
- Agent Group
 - Create an agent group
 - Delete an agent group
 - Retrieve an agent group
 - Retrieve all agent groups
 - Update an agent group
- Agent Role
 - Retrieve an agent role
 - Retrieve all agent roles
- Announcement
 - Create an announcement
 - Delete an announcement
 - Retrieve an announcement
 - Retrieve all announcements
 - Update an announcement

- Asset Type
 - Create an asset type
 - Delete an asset type
 - Retrieve an asset type
 - Retrieve all asset types
 - Update an asset type
- Change
 - Create a change
 - Delete a change
 - Retrieve a change
 - Retrieve all changes
 - Update a change
- Department
 - Create a department
 - Delete a department
 - Retrieve a department
 - Retrieve all departments
 - Update a department
- Location
 - Create a location
 - Delete a location
 - Retrieve a location
 - Retrieve all locations
 - Update a location
- Problem
 - Create a problem
 - Delete a problem
 - Retrieve a problem
 - Retrieve all problems
 - Update a problem
- Product
 - Create a product
 - Delete a product
 - Retrieve a product
 - Retrieve all products
 - Update a product
- Release
 - Create a release
 - Delete a release
 - Retrieve a release
 - Retrieve all releases
 - Update a release
- Requester
 - Create a requester
 - Delete a requester
 - Retrieve a requester
 - Retrieve all requesters
 - Update a requester
- Requester Group
 - Create a requester group
 - Delete a requester group
 - Retrieve a requester group
 - Retrieve all requester groups
 - Update a requester group
- Software
 - Create a software application
 - Delete a software application
 - Retrieve a software application
 - Retrieve all software applications
 - Update a software application

- Ticket
 - Create a ticket
 - Delete a ticket
 - Retrieve a ticket
 - Retrieve all tickets
 - Update a ticket

Templates and examples

Freshworks CRM node

Use the Freshworks CRM node to automate work in Freshworks CRM, and integrate Freshworks CRM with other applications. n8n has built-in support for a wide range of Freshworks CRM features, including creating, updating, deleting, and retrieve, accounts, appointments, contacts, deals, notes, sales activity and more.

On this page, you'll find a list of operations the Freshworks CRM node supports and links to more resources.

Operations

- Account
 - Create an account
 - Delete an account
 - Retrieve an account
 - Retrieve all accounts
 - Update an account
- Appointment
 - Create an appointment
 - Delete an appointment
 - Retrieve an appointment
 - Retrieve all appointments
 - Update an appointment
- Contact
 - Create a contact
 - Delete a contact
 - Retrieve a contact
 - Retrieve all contacts
 - Update a contact
- Deal
 - Create a deal
 - Delete a deal
 - Retrieve a deal
 - Retrieve all deals
 - Update a deal
- Note
 - Create a note
 - Delete a note
 - Update a note
- Sales Activity
 - Retrieve a sales activity
 - Retrieve all sales activities
- Task
 - Create a task

- Delete a task
- Retrieve a task
- Retrieve all tasks
- Update a task

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

GetResponse node

Use the GetResponse node to automate work in GetResponse, and integrate GetResponse with other applications. n8n has built-in support for a wide range of GetResponse features, including creating, updating, deleting, and getting contacts.

On this page, you’ll find a list of operations the GetResponse node supports and links to more resources.

Operations

- Contact
 - Create a new contact
 - Delete a contact
 - Get a contact
 - Get all contacts
 - Update contact properties

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Ghost node

Use the Ghost node to automate work in Ghost, and integrate Ghost with other applications. n8n has built-in support for a wide range of Ghost features, including creating, updating, deleting, and getting posts for the Admin and content API.

On this page, you’ll find a list of operations the Ghost node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

Admin API

- **Post**
 - Create a post
 - Delete a post
 - Get a post
 - Get all posts
 - Update a post

Content API

- **Post**
 - Get a post
 - Get all posts

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

GitHub node

Use the GitHub node to automate work in GitHub, and integrate GitHub with other applications. n8n has built-in support for a wide range of GitHub features, including creating, updating, deleting, and editing files, repositories, issues, releases, and users.

On this page, you'll find a list of operations the GitHub node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- File
 - Create
 - Delete
 - Edit
 - Get
 - List
- Issue
 - Create
 - Create Comment
 - Edit
 - Get
 - Lock
- Organization
 - Get Repositories
- Release
 - Create
 - Delete
 - Get
 - Get Many
 - Update
- Repository
 - Get

- Get Issues
- Get License
- Get Profile
- Get Pull Requests
- List Popular Paths
- List Referrers
- Review
 - Create
 - Get
 - Get Many
 - Update
- User
 - Get Repositories
 - Invite
- Workflow
 - Disable
 - Dispatch
 - Enable
 - Get
 - Get Usage
 - List

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

GitLab node

Use the GitLab node to automate work in GitLab, and integrate GitLab with other applications. n8n has built-in support for a wide range of GitLab features, including creating, updating, deleting, and editing issues, repositories, releases and users.

On this page, you’ll find a list of operations the GitLab node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- File
 - Create
 - Delete
 - Edit
 - Get
 - List
- Issue
 - Create a new issue
 - Create a new comment on an issue
 - Edit an issue
 - Get the data of a single issue
 - Lock an issue
- Release
 - Create a new release

- Delete a new release
- Get a new release
- Get all releases
- Update a new release
- Repository
 - Get the data of a single repository
 - Returns issues of a repository
- User
 - Returns the repositories of a user

Templates and examples

Related resources

Refer to [GitLab's documentation](#) for more information about the service.

n8n provides a trigger node for GitLab. You can find the trigger node docs [here](#).

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Gmail node

Use the Gmail node to automate work in Gmail, and integrate Gmail with other applications. n8n has built-in support for a wide range of Gmail features, including creating, updating, deleting, and getting drafts, messages, labels, thread.

On this page, you'll find a list of operations the Gmail node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- **Draft**
 - **Create** a draft
 - **Delete** a draft
 - **Get** a draft
 - **Get Many** drafts
- **Label**
 - **Create** a label
 - **Delete** a label
 - **Get** a label
 - **Get Many** labels
- **Message**
 - **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
- **Thread**
 - [**Add Label**](#) to a thread
 - [**Delete**](#) a thread
 - [**Get**](#) a thread
 - [**Get Many**](#) threads
 - [**Remove Label**](#) from thread
 - [**Reply**](#) to a message
 - [**Trash**](#) a thread
 - [**Untrash**](#) a thread

Templates and examples

Related resources

Refer to Google's [Gmail API documentation](#) for detailed information about the API that this node integrates with.

n8n provides a trigger node for Gmail. You can find the trigger node docs [here](#).

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Common issues

For common errors or issues and suggested resolution steps, refer to [Common Issues](#).

Gmail node Draft Operations

Use the Draft operations to create, delete, or get a draft or list drafts in Gmail. Refer to the [Gmail node](#) for more information on the Gmail node itself.

Create a draft

Use this operation to create a new draft.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Draft**.
- **Operation:** Select **Create**.
- **Subject:** Enter the subject line.
- Select the **Email Type**. Choose from **Text** or **HTML**.
- **Message:** Enter the email message body.

Create draft options

Use these options to further refine the node's behavior:

- **Attachments:** Select **Add Attachment** to add an attachment. Enter the **Attachment Field Name (in Input)** to identify which field from the input node contains the attachment.
 - For multiple properties, enter a comma-separated list.
- **BCC:** Enter one or more email addresses for blind copy recipients. Separate multiple email addresses with a comma, for example jay@gatsby.com, jon@smith.com.
- **CC:** Enter one or more email addresses for carbon copy recipients. Separate multiple email addresses with a comma, for example jay@gatsby.com, jon@smith.com.
- **From Alias Name or ID:** Select an alias to send the draft from. This field populates based on the credential you selected in the parameters.
- **Send Replies To:** Enter an email address to set as the reply to address.
- **Thread ID:** If you want this draft attached to a thread, enter the ID for that thread.
- **To Email:** Enter one or more email addresses for recipients. Separate multiple email addresses with a comma, for example jay@gatsby.com, jon@smith.com.

Refer to the [Gmail API Method: users.drafts.create](#) documentation for more information.

Delete a draft

Use this operation to delete a draft.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Draft**.
- **Operation:** Select **Delete**.
- **Draft ID:** Enter the ID of the draft you wish to delete.

Refer to the [Gmail API Method: users.drafts.delete](#) documentation for more information.

Get a draft

Use this operation to get a single draft.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Draft**.
- **Operation:** Select **Get**.
- **Draft ID:** Enter the ID of the draft you wish to get information about.

Get draft options

Use these options to further refine the node's behavior:

- **Attachment Prefix:** Enter a prefix for the name of the binary property the node should write any attachments to. n8n adds an index starting with 0 to the prefix. For example, if you enter `attachment_` as the prefix, the first attachment saves to `attachment_0`.
- **Download Attachments:** Select whether the node should download the draft's attachments (turned on) or not (turned off).

Refer to the [Gmail API Method: users.drafts.get](#) documentation for more information.

Get Many drafts

Use this operation to get two or more drafts.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Draft**.
- **Operation:** Select **Get Many**.
- **Return All:** Choose whether the node returns all drafts (turned on) or only up to a set limit (turned off).
- **Limit:** Enter the maximum number of drafts to return. Only used if you've turned off **Return All**.

Get Many drafts options

Use these options to further refine the node's behavior:

- **Attachment Prefix:** Enter a prefix for the name of the binary property the node should write any attachments to. n8n adds an index starting with 0 to the prefix. For example, if you enter `attachment_` as the prefix, the first attachment saves to `attachment_0`.
- **Download Attachments:** Select whether the node should download the draft's attachments (turned on) or not (turned off).
- **Include Spam and Trash:** Select whether the node should get drafts in the Spam and Trash folders (turned on) or not (turned off).

Refer to the [Gmail API Method: users.drafts.list](#) documentation for more information.

Common issues

For common errors or issues and suggested resolution steps, refer to [Common Issues](#).

Gmail node Label Operations

Use the Label operations to create, delete, or get a label or list labels in Gmail. Refer to the [Gmail node](#) for more information on the Gmail node itself.

Create a label

Use this operation to create a new label.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Label**.
- **Operation:** Select **Create**.
- **Name:** Enter a display name for the label.

Create label options

Use these options to further refine the node's behavior:

- **Label List Visibility:** Sets the visibility of the label in the label list in the Gmail web interface. Choose from:
 - **Hide:** Don't show the label in the label list.
 - **Show** (default): Show the label in the label list.
 - **Show if Unread:** Show the label if there are any unread messages with that label.
- **Message List Visibility:** Sets the visibility of messages with this label in the message list in the Gmail web interface. Choose whether to **Show** or **Hide** messages with this label.

Refer to the [Gmail API Method: users.labels.create](#) documentation for more information.

Delete a label

Use this operation to delete an existing label.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Label**.
- **Operation:** Select **Delete**.
- **Label ID:** Enter the ID of the label you want to delete.

Refer to the [Gmail API Method: users.labels.delete](#) documentation for more information.

Get a label

Use this operation to get an existing label.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Label**.
- **Operation:** Select **Get**.
- **Label ID:** Enter the ID of the label you want to get.

Refer to the [Gmail API Method: users.labels.get](#) documentation for more information.

Get Many labels

Use this operation to get two or more labels.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Label**.
- **Operation:** Select **Get Many**.
- **Return All:** Choose whether the node returns all labels (turned on) or only up to a set limit (turned off).
- **Limit:** Enter the maximum number of labels to return. Only used if you've turned off **Return All**.

Refer to the [Gmail API Method: users.labels.list](#) documentation for more information.

Common issues

For common errors or issues and suggested resolution steps, refer to [Common Issues](#).

Gmail node Message Operations

Use the Message operations to send, reply to, delete, mark read or unread, add a label to, remove a label from, or get a message or get a list of messages in Gmail. Refer to the [Gmail node](#) for more information on the Gmail node itself.

Add Label to a message

Use this operation to add one or more labels to a message.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Message**.
- **Operation:** Select **Add Label**.
- **Message ID:** Enter the ID of the message you want to add the label to.
- **Label Names or IDs:** Select the Label names you want to add or enter an expression to specify IDs. The dropdown populates based on the **Credential** you selected.

Refer to the [Gmail API Method: users.messages.modify](#) documentation for more information.

Delete a message

Use this operation to immediately and permanently delete a message.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Message**.
- **Operation:** Select **Delete**.
- **Message ID:** Enter the ID of the message you want to delete.

Refer to the [Gmail API Method: users.messages.delete](#) documentation for more information.

Get a message

Use this operation to get a single message.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Message**.
- **Operation:** Select **Get**.
- **Message ID:** Enter the ID of the message you wish to retrieve.
- **Simplify:** Choose whether to return a simplified version of the response (turned on) or the raw data (turned off). Default is on.
 - This is the same as setting the format for the API call to metadata, which returns email message IDs, labels, and email headers, including: From, To, CC, BCC, and Subject.

Refer to the [Gmail API Method: users.messages.get](#) documentation for more information.

Get Many messages

Use this operation to get two or more messages.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Message**.
- **Operation:** Select **Get Many**.
- **Return All:** Choose whether the node returns all messages (turned on) or only up to a set limit (turned off).
- **Limit:** Enter the maximum number of messages to return. Only used if you've turned off **Return All**.
- **Simplify:** Choose whether to return a simplified version of the response (turned on) or the raw data (turned off). Default is on.
 - This is the same as setting the format for the API call to metadata, which returns email message IDs, labels, and email headers, including: From, To, CC, BCC, and Subject.

Get Many messages filters

Use these filters to further refine the node's behavior:

- **Include Spam and Trash:** Select whether the node should get messages in the Spam and Trash folders (turned on) or not (turned off).
- **Label Names or IDs:** Only return messages with the selected labels added to them. Select the Label names you want to apply or enter an expression to specify IDs. The dropdown populates based on the **Credential** you selected.

- **Search:** Enter Gmail search refine filters, like `from:`, to filter the messages returned. Refer to [Refine searches in Gmail](#) for more information.
- **Read Status:** Choose whether to receive **Unread and read emails**, **Unread emails only** (default), or **Read emails only**.
- **Received After:** Return only those emails received after the specified date and time. Use the date picker to select the day and time or enter an expression to set a date as a string in ISO format or a timestamp in milliseconds. Refer to [ISO 8601](#) for more information on formatting the string.
- **Received Before:** Return only those emails received before the specified date and time. Use the date picker to select the day and time or enter an expression to set a date as a string in ISO format or a timestamp in milliseconds. Refer to [ISO 8601](#) for more information on formatting the string.
- **Sender:** Enter an email or a part of a sender name to return messages from only that sender.

Refer to the [Gmail API Method: users.messages.list](#) documentation for more information.

Mark as Read

Use this operation to mark a message as read.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Message**.
- **Operation:** Select **Mark as Read**.
- **Message ID:** Enter the ID of the message you wish to mark as read.

Refer to the [Gmail API Method: users.messages.modify](#) documentation for more information.

Mark as Unread

Use this operation to mark a message as unread.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Message**.
- **Operation:** Select **Mark as Unread**.
- **Message ID:** Enter the ID of the message you wish to mark as unread.

Refer to the [Gmail API Method: users.messages.modify](#) documentation for more information.

Remove Label from a message

Use this operation to remove one or more labels from a message.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Message**.
- **Operation:** Select **Remove Label**.
- **Message ID:** Enter the ID of the message you want to remove the label from.
- **Label Names or IDs:** Select the Label names you want to remove or enter an expression to specify IDs. The dropdown populates based on the **Credential** you selected.

Refer to the [Gmail API Method: users.messages.modify](#) documentation for more information.

Reply to a message

Use this operation to send a message as a reply to an existing message.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Message**.
- **Operation:** Select **Reply**.
- **Message ID:** Enter the ID of the message you want to reply to.
- Select the **Email Type**. Choose from **Text** or **HTML**.
- **Message:** Enter the email message body.

Reply options

Use these options to further refine the node's behavior:

- **Append n8n attribution:** By default, the node appends the statement This email was sent automatically with n8n to the end of the email. To remove this statement, turn this option off.
- **Attachments:** Select **Add Attachment** to add an attachment. Enter the **Attachment Field Name (in Input)** to identify which field from the input node contains the attachment.
 - For multiple properties, enter a comma-separated list.
- **BCC:** Enter one or more email addresses for blind copy recipients. Separate multiple email addresses with a comma, for example jay@gatsby.com, jon@smith.com.
- **CC:** Enter one or more email addresses for carbon copy recipients. Separate multiple email addresses with a comma, for example jay@gatsby.com, jon@smith.com.
- **Sender Name:** Enter the name you want displayed in your recipients' email as the sender.
- **Reply to Sender Only:** Choose whether to reply all (turned off) or reply to the sender only (turned on).

Refer to the [Gmail API Method: users.messages.send](#) documentation for more information.

Send a message

Use this operation to send a message.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Message**.
- **Operation:** Select **Send**.
- **To:** Enter the email address you want the email sent to.
- **Subject:** Enter the subject line.
- Select the **Email Type**. Choose from **Text** or **HTML**.
- **Message:** Enter the email message body.

Send options

Use these options to further refine the node's behavior:

- **Append n8n attribution:** By default, the node appends the statement This email was sent automatically with n8n to the end of the email. To remove this statement, turn this option off.
- **Attachments:** Select **Add Attachment** to add an attachment. Enter the **Attachment Field Name (in Input)** to identify which field from the input node contains the attachment.
 - For multiple properties, enter a comma-separated list.
- **BCC:** Enter one or more email addresses for blind copy recipients. Separate multiple email addresses with a comma, for example jay@gatsby.com, jon@smith.com.
- **CC:** Enter one or more email addresses for carbon copy recipients. Separate multiple email addresses with a comma, for example jay@gatsby.com, jon@smith.com.
- **Sender Name:** Enter the name you want displayed in your recipients' email as the sender.
- **Send Replies To:** Enter an email address to set as the reply to address.
- **Reply to Sender Only:** Choose whether to reply all (turned off) or reply to the sender only (turned on).

Refer to the [Gmail API Method: users.messages.send](#) documentation for more information.

Send a message and wait for approval

Use this operation to send a message and wait for approval from the recipient before continuing the workflow execution.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Message**.
- **Operation:** Select **Send and Wait for Approval**.
- **To:** Enter the email address you want the email sent to.
- **Subject:** Enter the subject line.
- **Message:** Enter the email message body.

Send and wait for approval options

Use these options to further refine the node's behavior:

- **Type of Approval:** Choose **Approve Only** (default) to include only an approval button or **Approve and Disapprove** to also include a disapproval option.
- **Approve Button Label:** The label to use for the approval button

(Approve by default).

- **Approve Button Style:** Whether to style the approval button as a **Primary** (default) or **Secondary** button.
- **Disapprove Button Label:** The label to use for the disapproval button (**Decline** by default). Only visible when you set **Type of Approval** to **Approve and Disapprove**.
- **Disapprove Button Style:** Whether to style the disapproval button as a **Primary** or **Secondary** (default) button. Only visible when you set **Type of Approval** to **Approve and Disapprove**.

Refer to the [Gmail API Method: users.messages.send](#) documentation for more information.

Common issues

For common errors or issues and suggested resolution steps, refer to [Common Issues](#).

Gmail node Thread Operations

Use the Thread operations to delete, reply to, trash, untrash, add/remove labels, get one, or list threads. Refer to the [Gmail node](#) for more information on the Gmail node itself.

Add Label to a thread

Use this operation to create a new draft.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Thread**.
- **Operation:** Select **Add Label**.
- **Thread ID:** Enter the ID of the thread you want to add the label to.
- **Label Names or IDs:** Select the Label names you want to apply or enter an expression to specify IDs. The dropdown populates based on the **Credential** you selected.

Refer to the [Gmail API Method: users.threads.modify](#) documentation for more information.

Delete a thread

Use this operation to immediately and permanently delete a thread and all its messages.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Thread**.
- **Operation:** Select **Delete**.
- **Thread ID:** Enter the ID of the thread you want to delete.

Refer to the [Gmail API Method: users.threads.delete](#) documentation for more information.

Get a thread

Use this operation to get a single thread.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Thread**.
- **Operation:** Select **Get**.
- **Thread ID:** Enter the ID of the thread you wish to retrieve.
- **Simplify:** Choose whether to return a simplified version of the response (turned on) or the raw data (turned off). Default is on.
 - This is the same as setting the format for the API call to `metadata`, which returns email message IDs, labels, and email headers, including: From, To, CC, BCC, and Subject.

Get thread options

Use these options to further refine the node's behavior:

- **Return Only Messages:** Choose whether to return only thread messages (turned on).

Refer to the [Gmail API Method: users.threads.get](#) documentation for more information.

Get Many threads

Use this operation to get two or more threads.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Thread**.
- **Operation:** Select **Get Many**.
- **Return All:** Choose whether the node returns all threads (turned on) or only up to a set limit (turned off).
- **Limit:** Enter the maximum number of threads to return. Only used if you've turned off **Return All**.

Get Many threads filters

Use these filters to further refine the node's behavior:

- **Include Spam and Trash:** Select whether the node should get threads in the Spam and Trash folders (turned on) or not (turned off).
- **Label Names or IDs:** Only return threads with the selected labels added to them. Select the Label names you want to apply or enter an expression to specify IDs. The dropdown populates based on the **Credential** you selected.
- **Search:** Enter Gmail search refine filters, like `from:`, to filter the threads returned. Refer to [Refine searches in Gmail](#) for more information.

- **Read Status:** Choose whether to receive **Unread and read emails**, **Unread emails only** (default), or **Read emails only**.
- **Received After:** Return only those emails received after the specified date and time. Use the date picker to select the day and time or enter an expression to set a date as a string in ISO format or a timestamp in milliseconds. Refer to [ISO 8601](#) for more information on formatting the string.
- **Received Before:** Return only those emails received before the specified date and time. Use the date picker to select the day and time or enter an expression to set a date as a string in ISO format or a timestamp in milliseconds. Refer to [ISO 8601](#) for more information on formatting the string.

Refer to the [Gmail API Method: users.threads.list](#) documentation for more information.

Remove label from a thread

Use this operation to remove a label from a thread.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Thread**.
- **Operation:** Select **Remove Label**.
- **Thread ID:** Enter the ID of the thread you want to remove the label from.
- **Label Names or IDs:** Select the Label names you want to remove or enter an expression to specify their IDs. The dropdown populates based on the **Credential** you selected.

Refer to the [Gmail API Method: users.threads.modify](#) documentation for more information.

Reply to a message

Use this operation to reply to a message.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Thread**.
- **Operation:** Select **Reply**.
- **Thread ID:** Enter the ID of the thread you want to reply to.
- **Message Snippet or ID:** Select the Message you want to reply to or enter an expression to specify its ID. The dropdown populates based on the **Credential** you selected.
- Select the **Email Type**. Choose from **Text** or **HTML**.
- **Message:** Enter the email message body.

Reply options

Use these options to further refine the node's behavior:

- **Attachments:** Select **Add Attachment** to add an attachment. Enter the **Attachment Field Name (in Input)** to identify which field from the input node contains the attachment.

- For multiple properties, enter a comma-separated list.
- **BCC:** Enter one or more email addresses for blind copy recipients. Separate multiple email addresses with a comma, for example jay@gatsby.com, jon@smith.com.
- **CC:** Enter one or more email addresses for carbon copy recipients. Separate multiple email addresses with a comma, for example jay@gatsby.com, jon@smith.com.
- **Sender Name:** Enter the name you want displayed in your recipients' email as the sender.
- **Reply to Sender Only:** Choose whether to reply all (turned off) or reply to the sender only (turned on).

Refer to the [Gmail API Method: users.messages.send](#) documentation for more information.

Trash a thread

Use this operation to move a thread and all its messages to the trash.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Thread**.
- **Operation:** Select **Trash**.
- **Thread ID:** Enter the ID of the thread you want to move to the trash.

Refer to the [Gmail API Method: users.threads.trash](#) documentation for more information.

Untrash a thread

Use this operation to recover a thread and all its messages from the trash.

Enter these parameters:

- Select the **Credential to connect with** or create a new one.
- **Resource:** Select **Thread**.
- **Operation:** Select **Untrash**.
- **Thread ID:** Enter the ID of the thread you want to move to the trash.

Refer to the [Gmail API Method: users.threads.untrash](#) documentation for more information.

Common issues

For common errors or issues and suggested resolution steps, refer to [Common Issues](#).

Gmail node common issues

Here are some common errors and issues with the [Gmail node](#) and steps to resolve or troubleshoot them.

Remove the n8n attribution from sent messages

If you're using the node to [send a message](#) or [reply to a message](#), the node appends this statement to the end of the email:

| This email was sent automatically with n8n

To remove this attribution:

1. In the node's **Options** section, select **Add option**.
2. Select **Append n8n attribution**.
3. Turn the toggle off.

Refer to [Send options](#) and [Reply options](#) for more information.

Forbidden - perhaps check your credentials

This error displays next to certain dropdowns in the node, like the **Label Names or IDs** dropdown. The full text looks something like this:

There was a problem loading the parameter options from server:
"Forbidden - perhaps check your credentials?"

The error most often displays when you're using a Google Service Account as the credential and the credential doesn't have **Impersonate a User** turned on.

Refer to [Google Service Account: Finish your n8n credential](#) for more information.

401 unauthorized error

The full text of the error looks like this:

```
401 - {"error": "unauthorized_client", "error_description": "Client is unauthorized to retrieve access tokens using this method, or client not authorized for any of the scopes requested."}
```

This error occurs when there's an issue with the credential you're using and its scopes or permissions.

To resolve:

1. For [OAuth2](#) credentials, make sure you've enabled the Gmail API in **APIs & Services > Library**. Refer to [Google OAuth2 Single Service - Enable APIs](#) for more information.
2. For [Service Account](#) credentials:
 1. [Enable domain-wide delegation](#).
 2. Make sure you add the Gmail API as part of the domain-wide delegation configuration.

Bad request - please check your parameters

This error most often occurs if you enter a Message ID, Thread ID, or Label ID that doesn't exist.

Try a **Get** operation with the ID to confirm it exists.

Gong node

Use the Gong node to automate work in Gong and integrate Gong with other applications. n8n has built-in support for a wide range of Gong features, which includes getting one or more calls and users.

On this page, you'll find a list of operations the Gong node supports, and links to more resources.

Operations

- Call
 - Get
 - Get Many
- User
 - Get
 - Get Many

Templates and examples

Related resources

Refer to [Gong's documentation](#) for more information about the service.

-8<- “ snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Google Ads node

Use the Google Ads node to automate work in Google Ads, and integrate Google Ads with other applications. n8n has built-in support for a wide range of Google Ads features, including getting campaigns.

On this page, you'll find a list of operations the Google Ads node supports and links to more resources.

Operations

- Campaign
 - Get all campaigns

- Get a campaign

Templates and examples

Related resources

Refer to [Google Ads' documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Google Analytics node

Use the Google Analytics node to automate work in Google Analytics, and integrate Google Analytics with other applications. n8n has built-in support for a wide range of Google Analytics features, including returning reports and user activities.

On this page, you'll find a list of operations the Google Analytics node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Report
 - Get
- User Activity
 - Search

Templates and examples

Related resources

Refer to [Google Analytics' documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Google BigQuery node

Use the Google BigQuery node to automate work in Google BigQuery, and integrate Google BigQuery with other applications. n8n has built-in support for a wide range of Google BigQuery features, including creating, and retrieving records.

On this page, you'll find a list of operations the Google BigQuery node supports and links to more resources.

```
-8<- "_snippets/integrations/builtin/app-nodes/ai-tools.md"
```

Operations

- Execute Query
- Insert

Templates and examples

Related resources

Refer to [Google BigQuery's documentation](#) for more information about the service.

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

Google Books node

Use the Google Books node to automate work in Google Books, and integrate Google Books with other applications. n8n has built-in support for a wide range of Google Books features, including retrieving a specific bookshelf resource for the specified user, adding volume to a bookshelf, and getting volume.

On this page, you'll find a list of operations the Google Books node supports and links to more resources.

Operations

- Bookshelf
 - Retrieve a specific bookshelf resource for the specified user
 - Get all public bookshelf resource for the specified user
- Bookshelf Volume
 - Add a volume to a bookshelf
 - Clears all volumes from a bookshelf
 - Get all volumes in a specific bookshelf for the specified user
 - Moves a volume within a bookshelf
 - Removes a volume from a bookshelf
- Volume
 - Get a volume resource based on ID
 - Get all volumes filtered by query

Templates and examples

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

Google Business Profile node

Use the Google Business Profile node to automate work in Google Business Profile and integrate Google Business Profile with other applications. n8n has built-in support for a wide range of Google Business Profile features, which includes creating, updating, and deleting posts and reviews.

On this page, you'll find a list of operations the Google Business Profile node supports, and links to more resources.

Operations

- Post
 - Create
 - Delete
 - Get
 - Get Many
 - Update
- Review
 - Delete Reply
 - Get
 - Get Many
 - Reply

Templates and examples

Related resources

n8n provides a trigger node for Google Business Profile. You can find the trigger node docs [here](#).

Refer to [Google Business Profile's documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Google Calendar node

Use the Google Calendar node to automate work in Google Calendar, and integrate Google Calendar with other applications. n8n has built-in support for a wide range of Google Calendar features, including adding, retrieving, deleting and updating calendar events.

On this page, you'll find a list of operations the Google Calendar node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- **Calendar**
 - **Availability**: If a time-slot is available in a calendar
- **Event**
 - **Create**: Add an event to calendar
 - **Delete**: Delete an event
 - **Get**: Retrieve an event
 - **Get Many**: Retrieve all events from a calendar
 - **Update**: Update an event

Templates and examples

Related resources

n8n provides a trigger node for Google Calendar. You can find the trigger node docs [here](#).

Refer to [Google Calendar's documentation](#) for more information about the service.

[View example workflows and related content](#) on n8n's website.

Google Calendar Calendar operations

Use this operation to check availability in a calendar in Google Calendar. Refer to [Google Calendar](#) for more information on the Google Calendar node itself.

Availability

Use this operation to check if a time-slot is available in a calendar.

Enter these parameters:

- **Credential to connect with**: Create or select an existing [Google Calendar credentials](#).
- **Resource**: Select **Calendar**.
- **Operation**: Select **Availability**.
- **Calendar**: Choose a calendar you want to check against. Select **From list** to choose the title from the dropdown list or **By ID** to enter a calendar ID.
- **Start Time**: The start time for the time-slot you want to check. By default, uses an expression evaluating to the current time (`{{ $now }}`).
- **End Time**: The end time for the time-slot you want to check. By default, uses an expression evaluating to an hour from now (`$now.plus(1, 'hour')`).

Options

- **Output Format**: Select the format for the availability information:
 - **Availability**: Returns if there are already events overlapping

with the given time slot or not.

- **Booked Slots:** Returns the booked slots.
- **RAW:** Returns the RAW data from the API.
- **Timezone:** The timezone used in the response. By default, uses the n8n timezone.

Refer to the [Freebusy: query | Google Calendar](#) API documentation for more information.

Google Calendar Event operations

Use these operations to create, delete, get, and update events in Google Calendar. Refer to [Google Calendar](#) for more information on the Google Calendar node itself.

Create

Use this operation to add an event to a Google Calendar.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Calendar credentials](#).
- **Resource:** Select **Event**.
- **Operation:** Select **Create**.
- **Calendar:** Choose a calendar you want to add an event to. Select **From list** to choose the title from the dropdown list or **By ID** to enter a calendar ID.
- **Start Time:** The start time for the event. By default, uses an expression evaluating to the current time (`{{ $now }}`).
- **End Time:** The end time for the event. By default, this uses an expression evaluating to an hour from now (`{{ $now.plus(1, 'hour') }}`).
- **Use Default Reminders:** Whether to enable default reminders for the event according to the calendar configuration.

Options

- **All Day:** Whether the event is all day or not.
- **Attendees:** Attendees to invite to the event.
- **Color Name or ID:** The color of the event. Choose from the list or specify the ID using an expression.
- **Conference Data:** Creates a conference link (Hangouts, Meet, etc.) and attaches it to the event.
- **Description:** A description for the event.
- **Guests Can Invite Others:** Whether attendees other than the organizer can invite others to the event.
- **Guests Can Modify:** Whether attendees other than the organizer can modify the event.

- **Guests Can See Other Guests:** Whether attendees other than the organizer can see who the event's attendees are.
- **ID:** Opaque identifier of the event.
- **Location:** Geographic location of the event as free-form text.
- **Max Attendees:** The maximum number of attendees to include in the response. If there are more than the specified number of attendees, only returns the participant.
- **Repeat Frequency:** The repetition interval for recurring events.
- **Repeat How Many Times?**: The number of instances to create for recurring events.
- **Repeat Until:** The date at which recurring events should stop.
- **RRULE:** Recurrence rule. When set, ignores the Repeat Frequency, Repeat How Many Times, and Repeat Until parameters.
- **Send Updates:** Whether to send notifications about the creation of the new event.
- **Show Me As:** Whether the event blocks time on the calendar.
- **Summary:** The title of the event.

Refer to the [Events: insert | Google Calendar](#) API documentation for more information.

Delete

Use this operation to delete an event from a Google Calendar.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Calendar credentials](#).
- **Resource:** Select **Event**.
- **Operation:** Select **Delete**.
- **Calendar:** Choose a calendar you want to delete an event from. Select **From list** to choose the title from the dropdown list or **By ID** to enter a calendar ID.
- **Event ID:** The ID of the event to delete.

Options

- **Send Updates:** Whether to send notifications about the deletion of the event.

Refer to the [Events: delete | Google Calendar](#) API documentation for more information.

Get

Use this operation to retrieve an event from a Google Calendar.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Calendar credentials](#).
- **Resource:** Select **Event**.
- **Operation:** Select **Get**.
- **Calendar:** Choose a calendar you want to get an event from.
Select **From list** to choose the title from the dropdown list or **By ID** to enter a calendar ID.
- **Event ID:** The ID of the event to get.

Options

- **Max Attendees:** The maximum number of attendees to include in the response. If there are more than the specified number of attendees, only returns the participant.
- **Return Next Instance of Recurrent Event:** Whether to return the next instance of a recurring event instead of the event itself.
- **Timezone:** The timezone used in the response. By default, uses the n8n timezone.

Refer to the [Events: get | Google Calendar API documentation](#) for more information.

Get Many

Use this operation to retrieve more than one event from a Google Calendar.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Calendar credentials](#).
- **Resource:** Select **Event**.
- **Operation:** Select **Get Many**.
- **Calendar:** Choose a calendar you want to get an event from.
Select **From list** to choose the title from the dropdown list or **By ID** to enter a calendar ID.
- **Return All:** Whether to return all results or only up to a given limit.
- **Limit:** (When “Return All” isn’t selected) The maximum number of results to return.
- **After:** Retrieve events that occur after this time. At least part of the event must be after this time. By default, this uses an expression evaluating to the current time (`{{ $now }}`). Switch the field to “fixed” to select a date from a date widget.
- **Before:** Retrieve events that occur before this time. At least part of the event must be before this time. By default, this uses an expression evaluating to the current time plus a week (`$now.plus({ week: 1 })`). Switch the field to “fixed” to select a date from a date widget.

Options

- **Fields:** Specify the fields to return. By default, returns a set of commonly used fields predefined by Google. Use “*” to return all fields. You can find out more in [Google Calendar’s documentation](#)

[on working with partial resources](#).

- **iCalUID:** Specifies an event ID (in the iCalendar format) to include in the response.
- **Max Attendees:** The maximum number of attendees to include in the response. If there are more than the specified number of attendees, only returns the participant.
- **Order By:** The order to use for the events in the response.
- **Query:** Free text search terms to find events that match. This searches all fields except for extended properties.
- **Recurring Event Handling:** What to do for recurring events:
 - **All Occurrences:** Return all instances of the recurring event for the specified time range.
 - **First Occurrence:** Return the first event of a recurring event within the specified time range.
 - **Next Occurrence:** Return the next instance of a recurring event within the specified time range.
- **Show Deleted:** Whether to include deleted events (with status equal to “cancelled”) in the results.
- **Show Hidden Invitations:** Whether to include hidden invitations in the results.
- **Timezone:** The timezone used in the response. By default, uses the n8n timezone.
- **Updated Min:** The lower bounds for an event’s last modification time (as an [RFC 3339 timestamp](#))

Refer to the [Events: list | Google Calendar](#) API documentation for more information.

Update

Use this operation to update an event in a Google Calendar.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Calendar credentials](#).
- **Resource:** Select **Event**.
- **Operation:** Select **Update**.
- **Calendar:** Choose a calendar you want to add an event to. Select **From list** to choose the title from the dropdown list or **By ID** to enter a calendar ID.
- **Event ID:** The ID of the event to update.
- **Modify:** For recurring events, choose whether to update the recurring event or a specific instance of the recurring event.

- **Use Default Reminders:** Whether to enable default reminders for the event according to the calendar configuration.
- **Update Fields:** The fields of the event to update:
 - **All Day:** Whether the event is all day or not.
 - **Attendees:** Attendees to invite to the event. You can choose to either add attendees or replace the existing attendee list.
 - **Color Name or ID:** The color of the event. Choose from the list or specify the ID using an expression.
 - **Description:** A description for the event.
 - **End:** The end time of the event.
 - **Guests Can Invite Others:** Whether attendees other than the organizer can invite others to the event.
 - **Guests Can Modify:** Whether attendees other than the organizer can make changes to the event.
 - **Guests Can See Other Guests:** Whether attendees other than the organizer can see who the event's attendees are.
 - **ID:** Opaque identifier of the event.
 - **Location:** Geographic location of the event as free-form text.
 - **Max Attendees:** The maximum number of attendees to include in the response. If there are more than the specified number of attendees, only returns the participant.
 - **Repeat Frequency:** The repetition interval for recurring events.
 - **Repeat How Many Times?:** The number of instances to create for recurring events.
 - **Repeat Until:** The date at which recurring events should stop.
 - **RRULE:** Recurrence rule. When set, ignores the Repeat Frequency, Repeat How Many Times, and Repeat Until parameters.
 - **Send Updates:** Whether to send notifications about the creation of the new event.
 - **Show Me As:** Whether the event blocks time on the calendar.
 - **Start:** The start time of the event.
 - **Summary:** The title of the event.
 - **Visibility:** The visibility of the event:
 - **Confidential:** The event is private. This value is provided for compatibility.
 - **Default:** Uses the default visibility for events on the calendar.

- **Public:** The event is public and the event details are visible to all readers of the calendar.
- **Private:** The event is private and only event attendees may view event details.

Refer to the [Events: update | Google Calendar API](#) documentation for more information.

Google Chat node

Use the Google Chat node to automate work in Google Chat, and integrate Google Chat with other applications. n8n has built-in support for a wide range of Google Chat features, including getting membership and spaces, as well as creating and deleting messages.

On this page, you'll find a list of operations the Google Chat node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Member
 - Get a membership
 - Get all memberships in a space
- Message
 - Create a message
 - Delete a message
 - Get a message
 - Send and Wait for Response
 - Update a message
- Space
 - Get a space
 - Get all spaces the caller is a member of

-8<- “_snippets/integrations/builtin/send-and-wait-operation.md”

Templates and examples

Google Cloud Firestore node

Use the Google Cloud Firestore node to automate work in Google Cloud Firestore, and integrate Google Cloud Firestore with other applications. n8n has built-in support for a wide range of Google Cloud Firestore features, including creating, deleting, and getting documents.

On this page, you'll find a list of operations the Google Cloud Firestore node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Document
 - Create a document
 - Create/Update a document
 - Delete a document
 - Get a document
 - Get all documents from a collection
 - Runs a query against your documents
- Collection
 - Get all root collections

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Google Cloud Natural Language node

Use the Google Cloud Natural Language node to automate work in Google Cloud Natural Language, and integrate Google Cloud Natural Language with other applications. n8n has built-in support for a wide range of Google Cloud Natural Language features, including analyzing documents.

On this page, you’ll find a list of operations the Google Cloud Natural Language node supports and links to more resources.

Operations

- Document
 - Analyze Sentiment

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Google Cloud Realtime Database node

Use the Google Cloud Realtime Database node to automate work in Google Cloud Realtime Database, and integrate Google Cloud Realtime Database with other applications. n8n has built-in support for a wide range of Google Cloud Realtime Database features, including writing, deleting, getting, and appending databases.

On this page, you'll find a list of operations the Google Cloud Realtime Database node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Write data to a database
- Delete data from a database
- Get a record from a database
- Append to a list of data
- Update item on a database

Templates and examples

Google Cloud Storage node

Use the Google Cloud Storage node to automate work in Google Cloud Storage, and integrate Google Cloud Storage with other applications. n8n has built-in support for a wide range of Google Cloud Storage features, including creating, updating, deleting, and getting buckets and objects.

On this page, you'll find a list of operations the Google Cloud Storage node supports and links to more resources.

Operations

- Bucket
 - Create
 - Delete
 - Get
 - Get Many
 - Update
- Object
 - Create
 - Delete
 - Get
 - Get Many
 - Update

Templates and examples

Related resources

Refer to Google's [Cloud Storage API documentation](#) for detailed information about the API that this node integrates with.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Google Contacts node

Use the Google Contacts node to automate work in Google Contacts, and integrate Google Contacts with other applications. n8n has built-in support for a wide range of Google Contacts features, including creating, updating, retrieving, deleting, and getting contacts.

On this page, you'll find a list of operations the Google Contacts node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Contact
 - Create a contact
 - Delete a contact
 - Get a contact
 - Retrieve all contacts
 - Update a contact

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Google Docs node

Use the Google Docs node to automate work in Google Docs, and integrate Google Docs with other applications. n8n has built-in support for a wide range of Google Docs features, including creating, updating, and getting documents.

On this page, you'll find a list of operations the Google Docs node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Document
 - Create
 - Get
 - Update

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Google Drive node

Use the Google Drive node to automate work in Google Drive, and integrate Google Drive with other applications. n8n has built-in support for a wide range of Google Drive features, including creating, updating, listing, deleting, and getting drives, files, and folders.

On this page, you'll find a list of operations the Google Drive node supports and links to more resources.

Operations

- **File**
 - [Copy](#) a file
 - [Create from text](#)
 - [Delete](#) a file
 - [Download](#) a file
 - [Move](#) a file
 - [Share](#) a file
 - [Update](#) a file
 - [Upload](#) a file
- **File/Folder**
 - [Search](#) files and folders
- **Folder**
 - [Create](#) a folder
 - [Delete](#) a folder
 - [Share](#) a folder
- **Shared Drive**
 - [Create](#) a shared drive
 - [Delete](#) a shared drive
 - [Get](#) a shared drive
 - [Get Many](#) shared drives
 - [Update](#) a shared drive

Templates and examples

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Google Drive File operations

Use this operation to create, delete, change, and manage files in Google Drive. Refer to [Google Drive](#) for more information on the Google Drive node itself.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Copy a file

Use this operation to copy a file to a drive.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Drive credentials](#).
- **Resource:** Select **File**.
- **Operation:** Select **Copy**.
- **File:** Choose a file you want to copy.
 - Select **From list** to choose the title from the dropdown list, **By URL** to enter the URL of the file, or **By ID** to enter the fileId. You can find the fileId in a shareable Google Drive file URL: <https://docs.google.com/document/d/fileId/edit#gid=0>. In your Google Drive, select **Share > Copy link** to get the shareable file URL.
 - **File Name:** The name to use for the new copy of the file.
- **Copy In The Same Folder:** Choose whether to copy the file to the same folder. If disabled, set the following:
 - **Parent Drive:** Select **From list** to choose the drive from the dropdown list, **By URL** to enter the URL of the drive, or **By ID** to enter the driveId.
 - **Parent Folder:** Select **From list** to choose the folder from the dropdown list, **By URL** to enter the URL of the folder, or **By ID** to enter the folderId.
 - You can find the driveId and folderID by visiting the shared drive or folder in your browser and copying the last URL component:
<https://drive.google.com/drive/u/1/folders/driveId>.

Options

- **Copy Requires Writer Permissions:** Select whether to enable readers and commenters to copy, print, or download the new file.
- **Description:** A short description of the file.

Refer to the [Method: files.copy | Google Drive API documentation](#) for more information.

Create from text

Use this operation to create a new file in a drive from provided text.

Enter these parameters: - **Credential to connect with:** Create or select an existing [Google Drive credentials](#). - **Resource:** Select **File**. - **Operation:** Select **Create From Text**. - **File Content:** Enter the file content to use to create the new file. - **File Name:** The name to use for the new file. - **Parent Drive:** Select **From list** to choose the drive from the dropdown list, **By URL** to enter the URL of the drive, or **By ID** to enter the driveId. - **Parent Folder:** Select **From list** to choose the folder from the dropdown list, **By URL** to enter the URL of the folder, or **By ID** to enter the folderId.

You can find the driveId and folderID by visiting the shared drive or folder in your browser and copying the last URL component:
<https://drive.google.com/drive/u/1/folders/driveId>.

Options

- **APP Properties:** A bundle of arbitrary key-value pairs which are private to the requesting app.
- **Properties:** A bundle of arbitrary key-value pairs which are visible to all apps.
- **Keep Revision Forever:** Choose whether to set the `keepForever` field in the new head revision. This only applies to files with binary content. You can keep a maximum of 200 revisions, after which you must delete the pinned revisions.
- **OCR Language:** An [ISO 639-1](#) language code to help the OCR interpret the content during import.
- **Use Content As Indexable Text:** Choose whether to mark the uploaded content as indexable text.
- **Convert to Google Document:** Choose whether to create a Google Document instead of the default .txt format. You must enable the Google Docs API in the [Google API Console](#) for this to work.

Refer to the [Method: files.insert | Google Drive API](#) documentation for more information.

Delete a file

Use this operation to delete a file from a drive.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Drive credentials](#).
- **Resource:** Select **File**.
- **Operation:** Select **Delete**.
- **File:** Choose a file you want to delete.
 - Select **From list** to choose the title from the dropdown list, **By URL** to enter the URL of the file, or **By ID** to enter the fileId.
 - You can find the fileId in a shareable Google Drive file URL: <https://docs.google.com/document/d/fileId/edit#gid=0>. In your Google Drive, select **Share > Copy link** to get the shareable file URL.

Options

- **Delete Permanently:** Choose whether to delete the file now instead of moving it to the trash.

Refer to the [Method: files.delete | Google Drive API](#) documentation for more information.

Download a file

Use this operation to download a file from a drive.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Drive credentials](#).
- **Resource:** Select **File**.
- **Operation:** Select **Download**.
- **File:** Choose a file you want to download.
 - Select **From list** to choose the title from the dropdown list, **By URL** to enter the URL of the file, or **By ID** to enter the fileId.
 - You can find the fileId in a shareable Google Drive file URL: <https://docs.google.com/document/d/fileId/edit#gid=0>. In your Google Drive, select **Share > Copy link** to get the shareable file URL.

Options

- **Put Output File in Field:** Choose the field name to place the binary file contents to make it available to following nodes.
- **Google File Conversion:** Choose the formats to export as when downloading Google Files:
 - **Google Docs:** Choose the export format to use when downloading Google Docs files: **HTML**, **MS Word Document**, **Open Office Document**, **PDF**, **Rich Text (rtf)**, or **Text (txt)**.
 - **Google Drawings:** Choose the export format to use when downloading Google Drawing files: **JPEG**, **PDF**, **PNG**, or **SVG**.
 - **Google Slides:** Choose the export format to use when downloading Google Slides files: **MS PowerPoint**, **OpenOffice Presentation**, or **PDF**.
 - **Google Sheets:** Choose the export format to use when downloading Google Sheets files: **CSV**, **MS Excel**, **Open Office Sheet**, or **PDF**.
- **File Name:** The name to use for the downloaded file.

Refer to the [Method: files.get | Google Drive API documentation](#) for more information.

Move a file

Use this operation to move a file to a different location in a drive.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Drive credentials](#).
- **Resource:** Select **File**.
- **Operation:** Select **Move**.
- **File:** Choose a file you want to move.
 - Select **From list** to choose the title from the dropdown list, **By URL** to enter the URL of the file, or **By ID** to enter the fileId.
 - You can find the fileId in a shareable Google Drive file URL: <https://docs.google.com/document/d/fileId/edit#gid=0>. In your Google Drive, select **Share > Copy link** to get the shareable file URL.
- **Parent Drive:** Select **From list** to choose the drive from the dropdown list, **By URL** to enter the URL of the drive, or **By ID** to enter the driveId.
- **Parent Folder:** Select **From list** to choose the folder from the dropdown list, **By URL** to enter the URL of the folder, or **By ID** to enter the folderId.

You can find the `driveId` and `folderID` by visiting the shared drive or folder in your browser and copying the last URL component:
<https://drive.google.com/drive/u/1/folders/driveId>.

Refer to the [Method: parents.insert | Google Drive API documentation](#) for more information.

Share a file

Use this operation to add sharing permissions to a file.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Drive credentials](#).
- **Resource:** Select **File**.
- **Operation:** Select **Share**.
- **File:** Choose a file you want to share.
 - Select **From list** to choose the title from the dropdown list, **By URL** to enter the URL of the file, or **By ID** to enter the `fileId`.
 - You can find the `fileId` in a shareable Google Drive file URL: <https://docs.google.com/document/d/fileId/edit#gid=0>. In your Google Drive, select **Share > Copy link** to get the shareable file URL.
- **Permissions:** The permissions to add to the file:
 - **Role:** Select what users can do with the file. Can be one of **Commenter, File Organizer, Organizer, Owner, Reader, Writer**.
 - **Type:** Select the scope of the new permission:
 - **User:** Grant permission to a specific user, defined by entering their **Email Address**.
 - **Group:** Grant permission to a specific group, defined by entering its **Email Address**.
 - **Domain:** Grant permission to a complete domain, defined by the **Domain**.
 - **Anyone:** Grant permission to anyone. Can optionally **Allow File Discovery** to make the file discoverable through search.

Options

- **Email Message:** A plain text custom message to include in the notification email.
- **Move to New Owners Root:** Available when trying to transfer ownership while sharing an item not in a shared drive. When enabled, moves the file to the new owner's My Drive root folder.
- **Send Notification Email:** Whether to send a notification email when sharing to users or groups.
- **Transfer Ownership:** Whether to transfer ownership to the specified user and downgrade the current owner to writer permissions.
- **Use Domain Admin Access:** Whether to perform the action as a domain administrator.

Refer to the [REST Resources: files | Google Drive API documentation](#) for more information.

Update a file

Use this operation to update a file.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Drive credentials](#).
- **Resource:** Select **File**.
- **Operation:** Select **Update**.
- **File to Update:** Choose a file you want to update.
 - Select **From list** to choose the title from the dropdown list, **By URL** to enter the URL of the file, or **By ID** to enter the fileId.
 - You can find the fileId in a shareable Google Drive file URL: <https://docs.google.com/document/d/fileId/edit#gid=0>. In your Google Drive, select **Share > Copy link** to get the shareable file URL.
- **Change File Content:** Choose whether to send new binary data to replace the existing file content. If enabled, fill in the following:
 - **Input Data Field Name:** The name of the input field that contains the binary file data you wish to use.
- **New Updated File Name:** A new name for the file if you want to update the filename.

Options

- **APP Properties:** A bundle of arbitrary key-value pairs which are private to the requesting app.
- **Properties:** A bundle of arbitrary key-value pairs which are visible to all apps.
- **Keep Revision Forever:** Choose whether to set the keepForever field in the new head revision. This only applies to files with binary content. You can keep a maximum of 200 revisions, after which you must delete the pinned revisions.
- **OCR Language:** An [ISO 639-1](#) language code to help the OCR interpret the content during import.
- **Use Content As Indexable Text:** Choose whether to mark the uploaded content as indexable text.
- **Move to Trash:** Whether to move the file to the trash. Only possible for the file owner.
- **Return Fields:** Return metadata fields about the file. Can be one or more of the following: **[All]**, **explicitlyTrashed**, **exportLinks**, **hasThumbnail**, **iconLink**, **ID**, **Kind**, **mimeType**, **Name**, **Permissions**, **Shared**, **Spaces**, **Starred**, **thumbnailLink**, **Trashed**, **Version**, or **webViewLink**.

Refer to the [Method: files.update | Google Drive API documentation](#) for more information.

Upload a file

Use this operation to upload a file.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Drive credentials](#).
- **Resource:** Select **File**.
- **Operation:** Select **Upload**.
- **Input Data Field Name:** The name of the input field that contains the binary file data you wish to use.
- **File Name:** The name to use for the new file.
- **Parent Drive:** Select **From list** to choose the drive from the dropdown list, **By URL** to enter the URL of the drive, or **By ID** to enter the driveId.
- **Parent Folder:** Select **From list** to choose the folder from the dropdown list, **By URL** to enter the URL of the folder, or **By ID** to enter the folderId.

You can find the `driveId` and `folderID` by visiting the shared drive or folder in your browser and copying the last URL component:

<https://drive.google.com/drive/u/1/folders/driveId>.

Options

- **APP Properties:** A bundle of arbitrary key-value pairs which are private to the requesting app.
- **Properties:** A bundle of arbitrary key-value pairs which are visible to all apps.
- **Keep Revision Forever:** Choose whether to set the `keepForever` field in the new head revision. This only applies to files with binary content. You can keep a maximum of 200 revisions, after which you must delete the pinned revisions.
- **OCR Language:** An [ISO 639-1](#) language code to help the OCR interpret the content during import.
- **Use Content As Indexable Text:** Choose whether to mark the uploaded content as indexable text.
- **Simplify Output:** Choose whether to return a simplified version of the response instead of including all fields.

Refer to the [Method: files.insert | Google Drive API documentation](#) for more information.

Google Drive File and Folder operations

Use this operation to search for files and folders in Google Drive.
Refer to [Google Drive](#) for more information on the Google Drive node itself.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Search files and folders

Use this operation to search for files and folders in a drive.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Drive credentials](#).
- **Resource:** Select **File/Folder**.
- **Operation:** Select **Search**.
- **Search Method:** Choose how you want to search:
 - **Search File/Folder Name:** Fill out the **Search Query** with the name of the file or folder you want to search for. Returns files and folders that are partial matches for the query as well.
 - **Advanced Search:** Fill out the **Query String** to search for files and folders using [Google query string syntax](#).
- **Return All:** Choose whether to return all results or only up to a given limit.
- **Limit:** The maximum number of items to return when **Return All** is disabled.
- **Filter:** Choose whether to limit the scope of your search:
 - **Drive:** The drive you want to search in. By default, uses your personal “My Drive”. Select **From list** to choose the drive from the dropdown list, **By URL** to enter the URL of the drive, or **By ID** to enter the `driveId`.
 - You can find the `driveId` by visiting the shared drive in your browser and copying the last URL component:
<https://drive.google.com/drive/u/1/folders/driveId>.
 - **Folder:** The folder to search in. Select **From list** to choose the folder from the dropdown list, **By URL** to enter the URL of the folder, or **By ID** to enter the `folderId`.
 - You can find the `folderId` by visiting the shared folder in your browser and copying the last URL component:
<https://drive.google.com/drive/u/1/folders/folderId>.
 - **What to Search:** Whether to search for **Files and Folders**, **Files**, or **Folders**.
 - **Include Trashed Items:** Whether to also return items in the Drive’s trash.

Options

- **Fields:** Select the fields to return. Can be one or more of the following: **[All]**, **explicitlyTrashed**, **exportLinks**, **hasThumbnail**, **iconLink**, **ID**, **Kind**, **mimeType**, **Name**, **Permissions**, **Shared**, **Spaces**, **Starred**, **thumbnailLink**, **Trashed**, **Version**, or **webViewLink**.

Refer to the [Method: files.list | Google Drive API documentation](#) for more information.

Google Drive Folder operations

Use this operation to create, delete, and share folders in Google Drive. Refer to [Google Drive](#) for more information on the Google Drive node itself.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Create a folder

Use this operation to create a new folder in a drive.

Enter these parameters: - **Credential to connect with:** Create or select an existing [Google Drive credentials](#). - **Resource:** Select **Folder**. - **Operation:** Select **Create**. - **Folder Name:** The name to use for the new folder. - **Parent Drive:** Select **From list** to choose the drive from the dropdown list, **By URL** to enter the URL of the drive, or **By ID** to enter the `driveId`. - **Parent Folder:** Select **From list** to choose the folder from the dropdown list, **By URL** to enter the URL of the folder, or **By ID** to enter the `folderId`.

You can find the `driveId` and `folderID` by visiting the shared drive or folder in your browser and copying the last URL component:
<https://drive.google.com/drive/u/1/folders/driveId>.

Options

- **Simplify Output:** Choose whether to return a simplified version of the response instead of including all fields.
- **Folder Color:** The color of the folder as an RGB hex string.

Refer to the [Method: files.insert | Google Drive API documentation](#) for more information.

Delete a folder

Use this operation to delete a folder from a drive.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Drive credentials](#).
- **Resource:** Select **Folder**.
- **Operation:** Select **Delete**.
- **Folder:** Choose a folder you want to delete.
 - Select **From list** to choose the folder from the dropdown list, **By URL** to enter the URL of the folder, or **By ID** to enter the `folderId`.
 - You can find the `folderId` in a Google Drive folder URL:
<https://drive.google.com/drive/u/0/folders/folderID>.

Options

- **Delete Permanently:** Choose whether to delete the folder now instead of moving it to the trash.

Refer to the [Method: files.delete | Google Drive API documentation](#) for more information.

Share a folder

Use this operation to add sharing permissions to a folder.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Drive credentials](#).
- **Resource:** Select **Folder**.
- **Operation:** Select **Share**.
- **Folder:** Choose a file you want to move.
 - Select **From list** to choose the folder from the dropdown list, **By URL** to enter the URL of the folder, or **By ID** to enter the `folderId`.
 - You can find the `folderId` in a Google Drive folder URL:
<https://drive.google.com/drive/u/0/folders/folderID>.
- **Permissions:** The permissions to add to the folder:
 - **Role:** Select what users can do with the folder. Can be one of **Commenter, File Organizer, Organizer, Owner, Reader, Writer**.
 - **Type:** Select the scope of the new permission:
 - **User:** Grant permission to a specific user, defined by entering their **Email Address**.
 - **Group:** Grant permission to a specific group, defined by entering its **Email Address**.
 - **Domain:** Grant permission to a complete domain, defined by the **Domain**.
 - **Anyone:** Grant permission to anyone. Can optionally **Allow File Discovery** to make the file discoverable through search.

Options

- **Email Message:** A plain text custom message to include in the notification email.
- **Move to New Owners Root:** Available when trying to transfer ownership while sharing an item not in a shared drive. When enabled, moves the folder to the new owner's My Drive root folder.
- **Send Notification Email:** Whether to send a notification email when sharing to users or groups.
- **Transfer Ownership:** Whether to transfer ownership to the specified user and downgrade the current owner to writer permissions.
- **Use Domain Admin Access:** Whether to perform the action as a domain administrator.

Refer to the [REST Resources: files | Google Drive API documentation](#) for more information.

Google Drive Shared Drive operations

Use this operation to create, delete, get, and update shared drives in Google Drive. Refer to [Google Drive](#) for more information on the Google Drive node itself.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Create a shared drive

Use this operation to create a new shared drive.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Drive credentials](#).
- **Resource:** Select **Shared Drive**.
- **Operation:** Select **Create**.
- **Name:** The name to use for the new shared drive.

Options

- **Capabilities:** The capabilities to set for the new shared drive (see [REST Resources: drives | Google Drive](#) for more details):
 - **Can Add Children:** Whether the current user can add children to folders in this shared drive.
 - **Can Change Copy Requires Writer Permission Restriction:** Whether the current user can change the copyRequiresWriterPermission restriction on this shared drive.
 - **Can Change Domain Users Only Restriction:** Whether the current user can change the domainUsersOnly restriction on this shared drive.
 - **Can Change Drive Background:** Whether the current user can change the background on this shared drive.
 - **Can Change Drive Members Only Restriction:** Whether the current user can change the driveMembersOnly restriction on this shared drive.
 - **Can Comment:** Whether the current user can comment on files in this shared drive.
 - **Can Copy:** Whether the current user can copy files in this shared drive.
 - **Can Delete Children:** Whether the current user can delete children from folders in this shared drive.
 - **Can Delete Drive:** Whether the current user can delete this shared drive. This operation may still fail if there are items not in the trash in the shared drive.
 - **Can Download:** Whether the current user can download files from this shared drive.
 - **Can Edit:** Whether the current user can edit files from this shared drive.
 - **Can List Children:** Whether the current user can list the children of folders in this shared drive.
 - **Can Manage Members:** Whether the current user can add, remove, or change the role of members of this shared drive.
 - **Can Read Revisions:** Whether the current user can read the revisions resource of files in this shared drive.
 - **Can Rename Drive:** Whether the current user can rename this shared drive.
 - **Can Share:** Whether the current user can share files or folders in this shared drive.

- **Can Trash Children:** Whether the current user can trash children from folders in this shared drive.
- **Color RGB:** The color of this shared drive as an RGB hex string.
- **Hidden:** Whether to hide this shared drive in the default view.
- **Restrictions:** Restrictions to add to this shared drive (see [REST Resources: drives | Google Drive](#) for more details):
 - **Admin Managed Restrictions:** When enabled, restrictions here will override the similarly named fields to true for any file inside of this shared drive.
 - **Copy Requires Writer Permission:** Whether the options to copy, print, or download files inside this shared drive should be disabled for readers and commenters.
 - **Domain Users Only:** Whether to restrict access to this shared drive and items inside this shared drive to users of the domain to which this shared drive belongs.
 - **Drive Members Only:** Whether to restrict access to items inside this shared drive to its members.

Refer to the [Method: drives.insert | Google Drive API documentation](#) for more information.

Delete a shared drive

Use this operation to delete a shared drive.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Drive credentials](#).
- **Resource:** Select **Shared Drive**.
- **Operation:** Select **Delete**.
- **Shared Drive:** Choose the shared drive want to delete.
 - Select **From list** to choose the title from the dropdown list, **By URL** to enter the URL of the drive, or **By ID** to enter the `driveId`.
 - You can find the `driveId` in the URL for the shared Google Drive: <https://drive.google.com/drive/u/0/folders/driveID>.

Refer to the [Method: drives.delete | Google Drive API documentation](#) for more information.

Get a shared drive

Use this operation to get a shared drive.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Drive credentials](#).
- **Resource:** Select **Shared Drive**.
- **Operation:** Select **Get**.
- **Shared Drive:** Choose the shared drive want to get.
 - Select **From list** to choose the title from the dropdown list, **By URL** to enter the URL of the drive, or **By ID** to enter the `driveId`.
 - You can find the `driveId` in the URL for the shared Google Drive: <https://drive.google.com/drive/u/0/folders/driveID>.

Options

- **Use Domain Admin Access:** Whether to issue the request as a domain administrator. When enabled, grants the requester access if they're an administrator of the domain to which the shared drive belongs.

Refer to the [Method: drives.get | Google Drive API documentation](#) for more information.

Get many shared drives

Use this operation to get many shared drives.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Drive credentials](#).
- **Resource:** Select **Shared Drive**.
- **Operation:** Select **Get Many**.
- **Return All:** Choose whether to return all results or only up to a given limit.
- **Limit:** The maximum number of items to return when **Return All** is disabled.
- **Shared Drive:** Choose the shared drive want to get.
 - Select **From list** to choose the title from the dropdown list, **By URL** to enter the URL of the drive, or **By ID** to enter the `driveId`.
 - You can find the `driveId` in the URL for the shared Google Drive: <https://drive.google.com/drive/u/0/folders/driveID>.

Options

- **Query:** The query string to use to search for shared drives. See [Search for shared drives | Google Drive](#) for more information.
- **Use Domain Admin Access:** Whether to issue the request as a domain administrator. When enabled, grants the requester access if they're an administrator of the domain to which the shared drive belongs.

Refer to the [Method: drives.get | Google Drive API documentation](#) for more information.

Update a shared drive

Use this operation to update a shared drive.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Drive credentials](#).
- **Resource:** Select **Shared Drive**.
- **Operation:** Select **Update**.
- **Shared Drive:** Choose the shared drive you want to update.
 - Select **From list** to choose the drive from the dropdown list, **By URL** to enter the URL of the drive, or **By ID** to enter the `driveId`.

- You can find the `driveId` in the URL for the shared Google Drive: <https://drive.google.com/drive/u/0/folders/driveID>.

Update Fields

- **Color RGB:** The color of this shared drive as an RGB hex string.
- **Name:** The updated name for the shared drive.
- **Restrictions:** Restrictions for this shared drive (see [REST Resources: drives | Google Drive](#) for more details):
 - **Admin Managed Restrictions:** When enabled, restrictions here will override the similarly named fields to true for any file inside of this shared drive.
 - **Copy Requires Writer Permission:** Whether the options to copy, print, or download files inside this shared drive should be disabled for readers and commenters.
 - **Domain Users Only:** Whether to restrict access to this shared drive and items inside this shared drive to users of the domain to which this shared drive belongs.
 - **Drive Members Only:** Whether to restrict access to items inside this shared drive to its members.

Refer to the [Method: drives.update | Google Drive API documentation](#) for more information.

Google Drive node common issues

Here are some common errors and issues with the [Google Drive node](#) and steps to resolve or troubleshoot them.

Google hasn't verified this app

-8<- “_snippets/integrations/builtin/credentials/google/unverified-app.md”

Google Cloud app becoming unauthorized

-8<- “_snippets/integrations/builtin/credentials/google/app-becoming-unauthorized.md”

Google Drive OAuth error

If using the OAuth authentication method, you may see an error indicating that you can't sign in because the app doesn't meet Google's expectations for keeping apps secure.

Most often, the actual cause of this issue is that the URLs don't match between Google's OAuth configuration and n8n. To avoid this, start by reviewing any links included in Google's error message. This will contain details about the exact error that occurred.

If you are self-hosting n8n, check the n8n configuration items used to construct external URLs. Verify that the `N8N_EDITOR_BASE_URL` and `WEBHOOK_URL` environment variables use fully qualified domains.

Get recent files from Google Drive

To retrieve recent files from Google Drive, you need to sort files by modification time. To do this, you need to search for existing files and retrieve their modification times. Next you can sort the files to find the most recent file and use another Google Drive node target the file by ID.

The process looks like this:

1. Add a **Google Drive** node to your canvas.
 2. Select the **File/Folder** resource and the **Search** operation.
 3. Enable **Return All** to sort through all files.
 4. Set the **What to Search** filter to **Files**.
 5. In the **Options**, set the **Fields** to **All**.
 6. Connect a **Sort** node to the output of the **Google Drive** node.
 7. Choose **Simple** sort type.
 8. Enter `modifiedTime` as the **Field Name** in the **Fields To Sort By** section.
 9. Choose **Descending** sort order.
 10. Add a **Limit** node to the output of the **Sort** node.
 11. Set **Max Items** to **1** to keep the most recent file.
 12. Connect another **Google Drive** node to the output of the **Limit** node.
 13. Select **File** as the **Resource** and the operation of your choice.
 14. In the **File** selection, choose **By ID**.
 15. Select **Expression** and enter `{{ $json.id }}` as the expression.
-

Google Gemini node

Use the Google Gemini node to automate work in Google Gemini and integrate Google Gemini with other applications. n8n has built-in support for a wide range of Google Gemini features, including working with audio, videos, images, documents, and files to analyze, generate, and transcribe.

On this page, you'll find a list of operations the Google Gemini node supports, and links to more resources.

Operations

- Audio:
 - Analyze Audio: Take in audio and answer questions about it.
 - Transcribe a Recording: Transcribes audio into text.
- Document:
 - Analyze Document: Take in documents and answer questions about them.
- File:
 - Upload File: Upload a file to the Google Gemini API for later user.
- Image:
 - Analyze Image: Take in images and answer questions about them.
 - Generate an Image: Creates an image from a text prompt.
- Text:

- Message a Model: Create a completion with a Google Gemini model.
- Video:
 - Analyze Video: Take in videos and answer questions about them.
 - Generate a Video: Creates a video from a text prompt.
 - Download Video: Download a generated video from the Google Gemini API using a URL.

Templates and examples

Related resources

Refer to [Google Gemini's documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Google Perspective node

Use the Google Perspective node to automate work in Google Perspective, and integrate Google Perspective with other applications. n8n has built-in support for a wide range of Google Perspective features, including analyzing comments.

On this page, you'll find a list of operations the Google Perspective node supports and links to more resources.

Operations

- Analyze Comment

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Google Sheets

Use the Google Sheets node to automate work in Google Sheets, and integrate Google Sheets with other applications. n8n has built-in support for a wide range of Google Sheets features, including creating, updating, deleting, appending, removing and getting documents.

On this page, you'll find a list of operations the Google Sheets node supports and links to more resources.

Operations

- **Document**
 - Create a spreadsheet.
 - Delete a spreadsheet.
- **Sheet Within Document**
 - Append or Update Row: Append a new row, or update the current one if it already exists.
 - Append Row: Create a new row.
 - Clear all data from a sheet.
 - Create a new sheet.
 - Delete a sheet.
 - Delete Rows or Columns: Delete columns and rows from a sheet.
 - Get Row(s): Read all rows in a sheet.
 - Update Row: Update a row in a sheet.

Templates and examples

Related resources

Refer to [Google Sheet's API documentation](#) for more information about the service.

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Google Sheets Document operations

Use this operation to create or delete a Google spreadsheet from Google Sheets. Refer to [Google Sheets](#) for more information on the Google Sheets node itself.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Create a spreadsheet

Use this operation to create a new spreadsheet.

Enter these parameters:

- **Credential to connect with**: Create or select an existing [Google Sheets credentials](#).
- **Resource**: Select **Document**.
- **Operation**: Select **Create**.

- **Title:** Enter the title of the new spreadsheet you want to create.
- **Sheets:** Add the **Title(s)** of the sheet(s) you want to create within the spreadsheet.

Options

- **Locale:** Enter the locale of the spreadsheet. This affects formatting details such as functions, dates, and currency. Use one of the following formats:
 - en (639-1)
 - fil (639-2 if no 639-1 format exists)
 - en_US (combination of ISO language and country).
 - Refer to [List of ISO 639 language codes](#) and [List of ISO 3166 country codes](#) for language and country codes. Note that Google doesn't support all locales/languages.
- **Recalculation Interval:** Enter the desired recalculation interval for the spreadsheet functions. This affects how often NOW, TODAY, RAND, and RANDBETWEEN are updated. Select **On Change** for recalculating whenever there is a change in the spreadsheet, **Minute** for recalculating every minute, or **Hour** for recalculating every hour. Refer to [Set a spreadsheet's location & calculation settings](#) for more information about these options.

Refer to the [Method: spreadsheets.create | Google Sheets API](#) documentation for more information.

Delete a spreadsheet

Use this operation to delete an existing spreadsheet.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Sheets credentials](#).
- **Resource:** Select **Document**.
- **Operation:** Select **Delete**.
- **Document:** Choose a spreadsheet you want to delete.
 - Select **From list** to choose the 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>.

Refer to the [Method: files.delete | Google Drive API](#) documentation for more information.

Google Sheets Sheet Within Document operations

Use this operation to create, update, clear or delete a sheet in a Google spreadsheet from Google Sheets. Refer to [Google Sheets](#) for more information on the Google Sheets node itself.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Append or Update Row

Use this operation to update an existing row or add a new row at the end of the data if a matching entry isn't found in a sheet.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Sheets credentials](#).
- **Resource:** Select **Sheet Within Document**.
- **Operation:** Select **Append or Update Row**.
- **Document:** Choose a spreadsheet that contains the sheet you want to append or update row(s) to.
 - 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>.
 - You can find the sheetId in a Google Sheets URL:
https://docs.google.com/spreadsheets/d/aBC-123_xYz/edit#gid=sheetId.
- **Sheet:** Choose a sheet you want to append or update row(s) to.
 - 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.
- **Mapping Column Mode:**
 - **Map Each Column Manually:** Enter **Values to Send** for each column.
 - **Map Automatically:** n8n looks for incoming data that matches the columns in Google Sheets automatically. In this mode, make sure the incoming data fields are the same as the columns in Google Sheets. (Use an [Edit Fields](#) node before this node to change them if required.)
 - **Nothing:** Don't map any data.

Options

```
-8<- "_snippets/integrations/builtin/app-nodes/googlesheets/node-options.md"
```

Refer to the [Method: spreadsheets.values.update | Google Sheets API](#) documentation for more information.

Append Row

Use this operation to append a new row at the end of the data in a sheet.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Sheets credentials](#).
- **Resource:** Select **Sheet Within Document**.
- **Operation:** Select **Append Row**.
- **Document:** Choose a spreadsheet with the sheet you want to append a row to.
 - 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:** Choose a sheet you want to append a row to.
 - 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`.
 - **Mapping Column Mode:**
 - **Map Each Column Manually:** Select the **Column to Match On** when finding the rows to update. Enter **Values to Send** for each column.
 - **Map Automatically:** n8n looks for incoming data that matches the columns in Google Sheets automatically. In this mode, make sure the incoming data fields are the same as the columns in Google Sheets. (Use an Edit Fields node before this node to change them if required.)
 - **Nothing:** Don't map any data.

Options

-8<- “_snippets/integrations/builtin/app-nodes/googlesheets/node-options.md”

Refer to the [Method: spreadsheets.values.append | Google Sheets API](#) documentation for more information.

Clear a sheet

Use this operation to clear all data from a sheet.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Sheets credentials](#).
- **Resource:** Select **Sheet Within Document**.
- **Operation:** Select **Clear**.
- **Document:** Choose a spreadsheet with the sheet you want to clear data from.
 - 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:** Choose a sheet you want to clear data from.
 - 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`.
- **Clear:** Select what data you want cleared from the sheet.
 - **Whole Sheet:** Clear the entire sheet's data. Turn on **Keep**

- **First Row** to keep the first row of the sheet.
- **Specific Rows:** Clear data from specific rows. Also enter:
 - **Start Row Number:** Enter the first row number you want to clear.
 - **Number of Rows to Delete:** Enter the number of rows to clear. 1 clears data only the row in the **Start Row Number**.
- **Specific Columns:** Clear data from specific columns. Also enter:
 - **Start Column:** Enter the first column you want to clear using the letter notation.
 - **Number of Columns to Delete:** Enter the number of columns to clear. 1 clears data only in the **Start Column**.
- **Specific Range:** Enter the table range to clear data from, in A1 notation.

Refer to the [Method: spreadsheets.values.clear | Google Sheets API](#) documentation for more information.

Create a new sheet

Use this operation to create a new sheet.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Sheets credentials](#).
- **Resource:** Select **Sheet Within Document**.
- **Operation:** Select **Create**.
- **Document:** Choose a spreadsheet in which you want to create a new sheet.
 - 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>.
- **Title:** Enter the title for your new sheet.

Options

- **Hidden:** Turn on this option to keep the sheet hidden in the UI.
- **Right To Left:** Turn on this option to use RTL sheet instead of an LTR sheet.
- **Sheet ID:** Enter the ID of the sheet.
 - You can find the sheetId in a Google Sheets URL:
https://docs.google.com/spreadsheets/d/aBC-123_xYz/edit#gid=sheetId
- **Sheet Index:** By default, the new sheet is the last sheet in the spreadsheet. To override this behavior, enter the index you want the new sheet to use. When you add a sheet at a given index, Google increments the indices for all following sheets. Refer to [Sheets | SheetProperties](#) documentation for more information.
- **Tab Color:** Enter the color as hex code or use the color picker to set the color of the tab in the UI.

Refer to the [Method: spreadsheets.batchUpdate | Google Sheets API](#) documentation for more information.

Delete a sheet

Use this operation to permanently delete a sheet.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Sheets credentials](#).
- **Resource:** Select **Sheet Within Document**.
- **Operation:** Select **Delete**.
- **Document:** Choose a spreadsheet that contains the sheet you want to delete.
 - 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:** Choose the sheet you want to delete.
 - 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 name of the sheet.
 - You can find the sheetId in a Google Sheets URL:
https://docs.google.com/spreadsheets/d/aBC-123_xYz/edit#gid=sheetId.

Refer to the [Method: spreadsheets.batchUpdate | Google Sheets API](#) documentation for more information.

Delete Rows or Columns

Use this operation to delete rows or columns in a sheet.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Sheets credentials](#).
- **Resource:** Select **Sheet Within Document**.
- **Operation:** Select **Delete Rows or Columns**.
- **Document:** Choose a spreadsheet that contains the sheet you want to delete rows or columns from.
 - 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:** Choose the sheet in which you want to delete rows or columns.
 - 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 name of the sheet.
 - You can find the sheetId in a Google Sheets URL:
https://docs.google.com/spreadsheets/d/aBC-123_xYz/edit#gid=sheetId.
- **Start Row Number or Start Column:** Enter the row number or column letter to start deleting.
- **Number of Rows to Delete or Number of Columns to delete:**

Enter the number of rows or columns to delete.

Refer to the [Method: spreadsheets.batchUpdate | Google Sheets API](#) documentation for more information.

Get Row(s)

Use this operation to read one or more rows from a sheet.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Sheets credentials](#).
- **Resource:** Select **Sheet Within Document**.
- **Operation:** Select **Get Row(s)**.
- **Document:** Choose a spreadsheet that contains the sheet you want to get rows from.
 - 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:** Choose a sheet you want to read rows from.
 - 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 name of the sheet.
 - You can find the sheetId in a Google Sheets URL:
https://docs.google.com/spreadsheets/d/aBC-123_xYz/edit#gid=sheetId
- **Filters:** By default, the node returns all rows in the sheet. Set filters to return a limited set of results:
 - **Column:** Select the column in your sheet to search against.
 - **Value:** Enter a cell value to search for. You can drag input data parameters here. If your filter matches multiple rows, n8n returns the first result. If you want all matching rows:
 1. Under **Options**, select **Add Option > When Filter Has Multiple Matches**.
 2. Change **When Filter Has Multiple Matches** to **Return All Matches**.

Options

- **Data Location on Sheet:** Use this option to specify a data range. By default, n8n will detect the range automatically until the last row in the sheet.
- **Output Formatting:** Use this option to choose how n8n formats the data returned by Google Sheets.
 - **General Formatting:**
 - **Values (unformatted)** (default): n8n removes currency signs and other special formatting. Data type remains as number.
 - **Values (formatted)**: n8n displays the values as they appear in Google Sheets (for example, retaining commas or currency signs) by converting the data type from number to string.
 - **Formulas**: n8n returns the formula. It doesn't calculate the formula output. For example, if a cell B2 has the formula

=A2, n8n returns B2's value as =A2 (in text). Refer to [About date & time values | Google Sheets](#) for more information.

- **Date Formatting:** Refer to [DateTimeRenderOption | Google Sheets](#) for more information. - **Formatted Text** (default): As displayed in Google Sheets, which depends on the spreadsheet locale. For example 01/01/2024. - **Serial Number:** Number of days since December 30th 1899.
- **When Filter Has Multiple Matches:** Set to **Return All Matches** to get multiple matches. By default only the first result gets returned.

Refer to the [Method: spreadsheets.batchUpdate | Google Sheets API](#) documentation for more information.

Update Row

Use this operation to update existing row in a sheet. This operation only updates existing rows. To append rows when a matching entry isn't found in a sheet, use **Append or Update Row** operation instead.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Google Sheets credentials](#).
- **Resource:** Select **Sheet Within Document**.
- **Operation:** Select **Update Row**.
- **Document:** Choose a spreadsheet with the sheet you want to update.
 - 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>.
 - 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.
- **Sheet:** Choose a sheet you want to update.
 - 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.
- **Mapping Column Mode:**
 - **Map Each Column Manually:** Enter **Values to Send** for each column.
 - **Map Automatically:** n8n looks for incoming data that matches the columns in Google Sheets automatically. In this mode, make sure the incoming data fields are the same as the columns in Google Sheets. (Use an [Edit Fields](#) node before this node to change them if required.)
 - **Nothing:** Don't map any data.

Options

- **Cell Format:** Use this option to choose how to format the data in cells. Refer to [Google Sheets API | CellFormat](#) for more information.
 - **Let Google Sheets format** (default): n8n formats text and numbers in the cells according to Google Sheets' default

- settings.
- **Let n8n format:** New cells in your sheet will have the same data types as the input data provided by n8n.
 - **Data Location on Sheet:** Use this option when you need to specify where the data range on your sheet.
 - **Header Row:** Specify the row index that contains the column headers.
 - **First Data Row:** Specify the row index where the actual data starts.

Refer to the [Method: spreadsheets.batchUpdate | Google Sheets API](#) documentation for more information.

Google Sheets node common issues

Here are some common errors and issues with the [Google Sheets node](#) and steps to resolve or troubleshoot them.

Append an array

To insert an array of data into Google Sheets, you must convert the array into a valid JSON (key, value) format.

To do so, consider using:

1. The [Split Out](#) node.
2. The [AI Transform](#) node. For example, try entering something like:
Convert 'languages' array to JSON (key, value) pairs.
3. The [Code](#) node.

Column names were updated after the node's setup

You'll receive this error if the Google Sheet's column names have changed since you set up the node.

To refresh the column names, re-select **Mapping Column Mode**. This should prompt the node to fetch the column names again.

Once the column names refresh, update the node parameters.

Google Slides node

Use the Google Slides node to automate work in Google Slides, and integrate Google Slides with other applications. n8n has built-in support for a wide range of Google Slides features, including creating presentations, and getting pages.

On this page, you'll find a list of operations the Google Slides node supports and links to more resources.

```
-8<- "_snippets/integrations/builtin/app-nodes/ai-tools.md"
```

Operations

- Page
 - Get a page
 - Get a thumbnail
- Presentation
 - Create a presentation
 - Get a presentation
 - Get presentation slides
 - Replace text in a presentation

Templates and examples

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

Google Tasks node

Use the Google Tasks node to automate work in Google Tasks, and integrate Google Tasks with other applications. n8n has built-in support for a wide range of Google Tasks features, including adding, updating, and retrieving contacts.

On this page, you'll find a list of operations the Google Tasks node supports and links to more resources.

```
-8<- "_snippets/integrations/builtin/app-nodes/ai-tools.md"
```

Operations

- Task
 - Add a task to task list
 - Delete a task
 - Retrieve a task
 - Retrieve all tasks from a task list
 - Update a task

Templates and examples

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

Google Translate node

Use the Google Translate node to automate work in Google Translate, and integrate Google Translate with other applications. n8n has built-in support for a wide range of Google Translate features, including translating languages.

On this page, you'll find a list of operations the Google Translate node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Language
 - Translate data

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Google Workspace Admin node

Use the Google Workspace Admin node to automate work in Google Workspace Admin, and integrate Google Workspace Admin with other applications. n8n has built-in support for a wide range of Google Workspace Admin features, including creating, updating, deleting, and getting users, groups, and ChromeOS devices.

On this page, you'll find a list of operations the Google Workspace Admin node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- ChromeOS Device
 - Get a ChromeOS device
 - Get many ChromeOS devices
 - Update a ChromeOS device
 - Change the status of a ChromeOS device
- Group
 - Create a group
 - Delete a group
 - Get a group
 - Get many groups
 - Update a group
- User
 - Add an existing user to a group
 - Create a user
 - Delete a user
 - Get a user
 - Get many users
 - Remove a user from a group
 - Update a user

Templates and examples

How to control which custom fields to fetch for a user

There are three different ways to control which custom fields to retrieve when getting a user's information. Use the **Custom Fields** parameter to select one of the following:

- **Don't Include**: Doesn't include any custom fields.
- **Custom**: Includes the custom fields from schemas in **Custom Schema Names or IDs**.
- **Include All**: Include all the fields associated with the user.

To include custom fields, follow these steps:

1. Select **Custom** from the **Custom Fields** dropdown list.
 2. Select the schema names you want to include in the **Custom Schema Names or IDs** dropdown list.
-

Gotify node

Use the Gotify node to automate work in Gotify, and integrate Gotify with other applications. n8n has built-in support for a wide range of Gotify features, including creating, deleting, and getting messages.

On this page, you'll find a list of operations the Gotify node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Message
 - Create
 - Delete
 - Get All

Templates and examples

GoToWebinar node

Use the GoToWebinar node to automate work in GoToWebinar, and integrate GoToWebinar with other applications. n8n has built-in support for a wide range of GoToWebinar features, including creating, getting, and deleting attendees, organizers, and registrants.

On this page, you'll find a list of operations the GoToWebinar node supports and links to more resources.

Operations

- Attendee
 - Get
 - Get All
 - Get Details
- Co-Organizer
 - Create
 - Delete
 - Get All
 - Re-invite
- Panelist
 - Create
 - Delete
 - Get All
 - Re-invite
- Registrant
 - Create
 - Delete
 - Get
 - Get All
- Session
 - Get
 - Get All
 - Get Details
- Webinar
 - Create
 - Get
 - Get All
 - Update

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Grafana node

Use the Grafana node to automate work in Grafana, and integrate Grafana with other applications. n8n has built-in support for a wide range of Grafana features, including creating, updating, deleting, and getting dashboards, teams, and users.

On this page, you’ll find a list of operations the Grafana node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Dashboard
 - Create a dashboard
 - Delete a dashboard
 - Get a dashboard
 - Get all dashboards
 - Update a dashboard
- Team

- Create a team
- Delete a team
- Get a team
- Retrieve all teams
- Update a team
- Team Member
 - Add a member to a team
 - Retrieve all team members
 - Remove a member from a team
- User
 - Delete a user from the current organization
 - Retrieve all users in the current organization
 - Update a user in the current organization

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Grist node

Use the Grist node to automate work in Grist, and integrate Grist with other applications. n8n has built-in support for a wide range of Grist features, including creating, updating, deleting, and reading rows in a table.

On this page, you’ll find a list of operations the Grist node supports and links to more resources.

Operations

- Create rows in a table
- Delete rows from a table
- Read rows from a table
- Update rows in a table

Templates and examples

Get the Row ID

To update or delete a particular record, you need the Row ID. There are two ways to get the Row ID:

Create a Row ID column in Grist

Create a new column in your Grist table with the formula `$id`.

Use the Get All operation

The **Get All** operation returns the Row ID of each record along with the fields.

You can get it with the expression
`>{{$node["GristNodeName"] . json["id"]}}`.

Filter records when using the Get All operation

- Select **Add Option** and select **Filter** from the dropdown list.
 - You can add filters for any number of columns. The result will only include records which match all the columns.
 - For each column, you can enter any number of values separated by commas. The result will include records which match any of the values for that column.
-

Hacker News node

Use the Hacker News node to automate work in Hacker News, and integrate Hacker News with other applications. n8n has built-in support for a wide range of Hacker News features, including getting articles, and users.

On this page, you'll find a list of operations the Hacker News node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- All
 - Get all items
- Article
 - Get a Hacker News article
- User
 - Get a Hacker News user

Templates and examples

HaloPSA node

Use the HaloPSA node to automate work in HaloPSA, and integrate HaloPSA with other applications. n8n has built-in support for a wide range of HaloPSA features, including creating, updating, deleting, and getting clients, sites and tickets.

On this page, you'll find a list of operations the HaloPSA node supports and links to more resources.

Operations

- Client

- Create a client
- Delete a client
- Get a client
- Get all clients
- Update a client
- Site
 - Create a site
 - Delete a site
 - Get a site
 - Get all sites
 - Update a site
- Ticket
 - Create a ticket
 - Delete a ticket
 - Get a ticket
 - Get all tickets
 - Update a ticket
- User
 - Create a user
 - Delete a user
 - Get a user
 - Get all users
 - Update a user

Templates and examples

Harvest node

Use the Harvest node to automate work in Harvest, and integrate Harvest with other applications. n8n has built-in support for a wide range of Harvest features, including creating, updating, deleting, and getting clients, contacts, invoices, tasks, expenses, users, and projects.

On this page, you'll find a list of operations the Harvest node supports and links to more resources.

Operations

- Client
 - Create a client
 - Delete a client
 - Get data of a client
 - Get data of all clients
 - Update a client
- Company
 - Retrieves the company for the currently authenticated user
- Contact
 - Create a contact
 - Delete a contact
 - Get data of a contact
 - Get data of all contacts
 - Update a contact
- Estimate
 - Create an estimate

- Delete an estimate
- Get data of an estimate
- Get data of all estimates
- Update an estimate
- Expense
 - Get data of an expense
 - Get data of all expenses
 - Create an expense
 - Update an expense
 - Delete an expense
- Invoice
 - Get data of an invoice
 - Get data of all invoices
 - Create an invoice
 - Update an invoice
 - Delete an invoice
- Project
 - Create a project
 - Delete a project
 - Get data of a project
 - Get data of all projects
 - Update a project
- Task
 - Create a task
 - Delete a task
 - Get data of a task
 - Get data of all tasks
 - Update a task
- Time Entries
 - Create a time entry using duration
 - Create a time entry using start and end time
 - Delete a time entry
 - Delete a time entry's external reference.
 - Get data of a time entry
 - Get data of all time entries
 - Restart a time entry
 - Stop a time entry
 - Update a time entry
- User
 - Create a user
 - Delete a user
 - Get data of a user
 - Get data of all users
 - Get data of authenticated user
 - Update a user

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Help Scout node

Use the Help Scout node to automate work in Help Scout, and integrate Help Scout with other applications. n8n has built-in support for a wide range of Help Scout features, including creating, updating,

deleting, and getting conversations, and customers.

On this page, you'll find a list of operations the Help Scout node supports and links to more resources.

Operations

- Conversation
 - Create a new conversation
 - Delete a conversation
 - Get a conversation
 - Get all conversations
- Customer
 - Create a new customer
 - Get a customer
 - Get all customers
 - Get customer property definitions
 - Update a customer
- Mailbox
 - Get data of a mailbox
 - Get all mailboxes
- Thread
 - Create a new chat thread
 - Get all chat threads

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

HighLevel node

Use the HighLevel node to automate work in HighLevel, and integrate HighLevel with other applications. n8n has built-in support for a wide range of HighLevel features, including creating, updating, deleting, and getting contacts, opportunities, and tasks, as well as booking appointments and getting free time slots in calendars.

On this page, you'll find a list of operations the HighLevel node supports and links to more resources.

Operations

- Contact
 - Create or update
 - Delete
 - Get
 - Get many
 - Update
- Opportunity
 - Create
 - Delete
 - Get
 - Get many

- Update
- Task
 - Create
 - Delete
 - Get
 - Get many
 - Update
- Calendar
 - Book an appointment
 - Get free slots

Templates and examples

Related resources

Refer to [HighLevel's API documentation and support forums](#) for more information about the service.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Home Assistant node

Use the Home Assistant node to automate work in Home Assistant, and integrate Home Assistant with other applications. n8n has built-in support for a wide range of Home Assistant features, including getting, creating, and checking camera proxies, configurations, logs, services, and templates.

On this page, you'll find a list of operations the Home Assistant node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Camera Proxy
 - Get the camera screenshot
- Config
 - Get the configuration
 - Check the configuration
- Event
 - Create an event
 - Get all events
- Log
 - Get a log for a specific entity
 - Get all logs
- Service
 - Call a service within a specific domain
 - Get all services
- State
 - Create a new record, or update the current one if it already exists (upsert)
 - Get a state for a specific entity

- Get all states
- Template
 - Create a template

Templates and examples

Related resources

Refer to [Home Assistant's documentation](#) for more information about the service.

HubSpot node

Use the HubSpot node to automate work in HubSpot, and integrate HubSpot with other applications. n8n has built-in support for a wide range of HubSpot features, including creating, updating, deleting, and getting contacts, deals, lists, engagements and companies.

On this page, you'll find a list of operations the HubSpot node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Contact
 - Create/Update a contact
 - Delete a contact
 - Get a contact
 - Get all contacts
 - Get recently created/updated contacts
 - Search contacts
- Contact List
 - Add contact to a list
 - Remove a contact from a list
- Company
 - Create a company
 - Delete a company
 - Get a company
 - Get all companies
 - Get recently created companies
 - Get recently modified companies
 - Search companies by domain
 - Update a company
- Deal
 - Create a deal
 - Delete a deal
 - Get a deal
 - Get all deals
 - Get recently created deals
 - Get recently modified deals
 - Search deals
 - Update a deal
- Engagement

- Create an engagement
- Delete an engagement
- Get an engagement
- Get all engagements
- Form
 - Get all fields from a form
 - Submit data to a form
- Ticket
 - Create a ticket
 - Delete a ticket
 - Get a ticket
 - Get all tickets
 - Update a ticket

Templates and examples

-8<- “[snippets/integrations/builtin/app-nodes/operation-not-supported.md](#)”

Humantic AI node

Use the Humantic AI node to automate work in Humantic AI, and integrate Humantic AI with other applications. n8n has built-in support for a wide range of Humantic AI features, including creating, retrieving, and updating profiles.

On this page, you’ll find a list of operations the Humantic AI node supports and links to more resources.

Operations

- Profile
 - Create a profile
 - Retrieve a profile
 - Update a profile

Templates and examples

Hunter node

Use the Hunter node to automate work in Hunter, and integrate Hunter with other applications. n8n has built-in support for a wide range of Hunter features, including getting, generating, and verifying email addresses.

On this page, you’ll find a list of operations the Hunter node supports and links to more resources.

Operations

- Get every email address found on the internet using a given domain name, with sources
- Generate or retrieve the most likely email address from a domain name, a first name and a last name
- Verify the deliverability of an email address

Templates and examples

Intercom node

Use the Intercom node to automate work in Intercom, and integrate Intercom with other applications. n8n has built-in support for a wide range of Intercom features, including creating, updating, deleting, and getting companies, leads, and users.

On this page, you'll find a list of operations the Intercom node supports and links to more resources.

Operations

- Company
 - Create a new company
 - Get data of a company
 - Get data of all companies
 - Update a company
 - List company's users
- Lead
 - Create a new lead
 - Delete a lead
 - Get data of a lead
 - Get data of all leads
 - Update new lead
- User
 - Create a new user
 - Delete a user
 - Get data of a user
 - Get data of all users
 - Update a user

Templates and examples

Invoice Ninja node

Use the Invoice Ninja node to automate work in Invoice Ninja, and integrate Invoice Ninja with other applications. n8n has built-in support for a wide range of Invoice Ninja features, including creating, updating, deleting, and getting clients, expense, invoice, payments and quotes.

On this page, you'll find a list of operations the Invoice Ninja node supports and links to more resources.

Operations

- Client
 - Create a new client
 - Delete a client
 - Get data of a client
 - Get data of all clients
- Expense
 - Create a new expense
 - Delete an expense
 - Get data of an expense
 - Get data of all expenses
- Invoice
 - Create a new invoice
 - Delete a invoice
 - Email an invoice
 - Get data of a invoice
 - Get data of all invoices
- Payment
 - Create a new payment
 - Delete a payment
 - Get data of a payment
 - Get data of all payments
- Quote
 - Create a new quote
 - Delete a quote
 - Email an quote
 - Get data of a quote
 - Get data of all quotes
- Task
 - Create a new task
 - Delete a task
 - Get data of a task
 - Get data of all tasks

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Iterable node

Use the Iterable node to automate work in Iterable, and integrate Iterable with other applications. n8n has built-in support for a wide range of Iterable features, including creating users, recording the actions performed by the users, and adding and removing users from the list.

On this page, you'll find a list of operations the Iterable node supports and links to more resources.

Operations

- Event

- Record the actions a user perform
- User
 - Create/Update a user
 - Delete a user
 - Get a user
- User List
 - Add user to list
 - Remove a user from a list

Templates and examples

Jenkins node

Use the Jenkins node to automate work in Jenkins, and integrate Jenkins with other applications. n8n has built-in support for a wide range of Jenkins features, including listing builds, managing instances, and creating and copying jobs.

On this page, you'll find a list of operations the Jenkins node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Build
 - List Builds
- Instance
 - Cancel quiet down state
 - Put Jenkins in quiet mode, no builds can be started, Jenkins is ready for shutdown
 - Restart Jenkins immediately on environments where it's possible
 - Restart Jenkins once no jobs are running on environments where it's possible
 - Shutdown once no jobs are running
 - Shutdown Jenkins immediately
- Job
 - Copy a specific job
 - Create a new job
 - Trigger a specific job
 - Trigger a specific job

Templates and examples

Jina AI node

Use the Jina AI node to automate work in Jina AI and integrate Jina AI with other applications. n8n has built-in support for a wide range of Jina AI features.

On this page, you'll find a list of operations the Jina AI node supports, and links to more resources.

Operations

- **Reader:**
 - **Read:** Fetches content from a URL and converts it to clean, LLM-friendly formats.
 - **Search:** Performs a web search using Jina AI and returns the top results as clean, LLM-friendly formats.
- **Research:**
 - **Deep Research:** Research a topic and generate a structured research report.

Templates and examples

Related resources

Refer to [Jina AI's reader API documentation](#) and [Jina AI's search API documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Jira Software node

Use the Jira Software node to automate work in Jira, and integrate Jira with other applications. n8n has built-in support for a wide range of Jira features, including creating, updating, deleting, and getting issues, and users.

On this page, you'll find a list of operations the Jira Software node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Issue
 - Get issue changelog
 - Create a new issue
 - Delete an issue
 - Get an issue
 - Get all issues
 - Create an email notification for an issue and add it to the mail queue
 - Return either all transitions or a transition that can be performed by the user on an issue, based on the issue's status
 - Update an issue
- Issue Attachment
 - Add attachment to issue
 - Get an attachment

- Get all attachments
- Remove an attachment
- Issue Comment
 - Add comment to issue
 - Get a comment
 - Get all comments
 - Remove a comment
 - Update a comment
- User
 - Create a new user.
 - Delete a user.
 - Retrieve a user.

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Related resources

Refer to the [official JQL documentation](#) about Jira Query Language (JQL) to learn more about it.

Fetch issues for a specific project

The **Get All** operation returns all the issues from Jira. To fetch issues for a particular project, you need to use Jira Query Language (JQL).

For example, if you want to receive all the issues of a project named n8n, you'd do something like this:

- Select **Get All** from the **Operation** dropdown list.
- Toggle **Return All** to true.
- Select **Add Option** and select **JQL**.
- Enter `project=n8n` in the **JQL** field.

This query will fetch all the issues in the project named n8n. Enter the name of your project instead of n8n to fetch all the issues for your project.

Kafka node

Use the Kafka node to automate work in Kafka, and integrate Kafka with other applications. n8n has built-in support for a wide range of Kafka features, including sending messages.

On this page, you'll find a list of operations the Kafka node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Send message

Templates and examples

Keap node

Use the Keap node to automate work in Keap, and integrate Keap with other applications. n8n has built-in support for a wide range of Keap features, including creating, updating, deleting, and getting companies, products, ecommerce orders, emails, and files.

On this page, you'll find a list of operations the Keap node supports and links to more resources.

Operations

- Company
 - Create a company
 - Retrieve all companies
- Contact
 - Create/update a contact
 - Delete an contact
 - Retrieve an contact
 - Retrieve all contacts
- Contact Note
 - Create a note
 - Delete a note
 - Get a notes
 - Retrieve all notes
 - Update a note
- Contact Tag
 - Add a list of tags to a contact
 - Delete a contact's tag
 - Retrieve all contact's tags
- Ecommerce Order
 - Create an ecommerce order
 - Get an ecommerce order
 - Delete an ecommerce order
 - Retrieve all ecommerce orders
- Ecommerce Product
 - Create an ecommerce product
 - Delete an ecommerce product
 - Get an ecommerce product
 - Retrieve all ecommerce product
- Email
 - Create a record of an email sent to a contact
 - Retrieve all sent emails
 - Send Email
- File
 - Delete a file
 - Retrieve all files
 - Upload a file

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Kitemaker node

Use the Kitemaker node to automate work in Kitemaker, and integrate Kitemaker with other applications. n8n has built-in support for a wide range of Kitemaker features, including retrieving data on organizations, spaces and users, as well as creating, getting, and updating work items.

On this page, you’ll find a list of operations the Kitemaker node supports and links to more resources.

Operations

- Organization
 - Retrieve data on the logged-in user’s organization.
- Space
 - Retrieve data on all the spaces in the logged-in user’s organization.
- User
 - Retrieve data on all the users in the logged-in user’s organization.
- Work Item
 - Create
 - Get
 - Get All
 - Update

Templates and examples

KoboToolbox node

Use the KoboToolbox node to automate work in KoboToolbox, and integrate KoboToolbox with other applications. n8n has built-in support for a wide range of KoboToolbox features, including creating, updating, deleting, and getting files, forms, hooks, and submissions.

On this page, you’ll find a list of operations the KoboToolbox node supports and links to more resources.

Operations

- File
 - Create
 - Delete
 - Get

- Get Many
- Form
 - Get
 - Get Many
 - Redeploy
- Hook
 - Get
 - Get Many
 - Logs
 - Retry All
 - Retry One
- Submission
 - Delete
 - Get
 - Get Many
 - Get Validation Status
 - Update Validation Status

Templates and examples

Options

Query Options

The Query Submission operation supports query options:

- In the main section of the **Parameters** panel:
 - **Start** controls the index offset to start the query from (to use the API pagination logic).
 - **Limit** sets the maximum number of records to return. Note that the API always has a limit of 30,000 returned records, whatever value you provide.
- In the **Query Options** section, you can activate the following parameters:
 - **Query** lets you specify filter predicates in MongoDB's JSON query format. For example: `{"status": "success", "_submission_time": {"$lt": "2021-11-01T01:02:03"}}` queries for all submissions with the value success for the field status, and submitted before November 1st, 2021, 01:02:03.
 - **Fields** lets you specify the list of fields you want to fetch, to make the response lighter.
 - **Sort** lets you provide a list of sorting criteria in MongoDB JSON format. For example, `{"status": 1, "_submission_time": -1}` specifies a sort order by ascending status, and then descending submission time.

More details about these options can be found in the [Formhub API docs](#)

Submission options

All operations that return form submission data offer options to tweak the response. These include:

- Download options lets you download any attachment linked to each particular form submissions, such as pictures and videos. It also lets you select the naming pattern, and the file size to download (if

available - typically for images).

- Formatting options perform some reformatting as described in [About reformatting](#).

About reformatting

The default JSON format for KoboToolbox submission data is sometimes hard to deal with, because it's not schema-aware, and all fields are therefore returned as strings.

This node provides a lightweight opinionated reformatting logic, enabled with the **Reformat?** parameter, available on all operations that return form submissions: the submission query, get, and the attachment download operations.

When enabled, the reformatting:

- Reorganizes the JSON into a multi-level hierarchy following the form's groups. By default, question grouping hierarchy is materialized by a / character in the field names, for example Group1/Question1. With reformatting enabled, n8n reorganizes these into Group1.Question1, as nested JSON objects.
- Renames fields to trim _ (not supported by many downstream systems).
- Parses all geospatial fields (Point, Line, and Area question types) into their standard GeoJSON equivalent.
- Splits all fields matching any of the **Multiselect Mask** wildcard masks into an array. Since the multi-select fields appear as space-separated strings, they can't be guessed algorithmically, so you must provide a field naming mask. Format the masks as a comma-separated list. Lists support the * wildcard.
- Converts all fields matching any of the **Number Mask** wildcard masks into a JSON float.

Here's a detailed example in JSON:

```
{
  "_id": 471987,
  "formhub/uuid": "189436bb09a54957bfcc798e338b54d6",
  "start": "2021-12-05T16:13:38.527+02:00",
  "end": "2021-12-05T16:15:33.407+02:00",
  "Field_Details/Field_Name": "Test Fields",
  "Field_Details/Field_Location": "-1.932914 30.078211 1421 165",
  "Field_Details/Field_Shape": "-1.932914 30.078211 1421
165;-1.933011 30.078085 0 0;-1.933257 30.078004 0 0;-1.933338
30.078197 0 0;-1.933107 30.078299 0 0;-1.932914 30.078211 1421 165",
  "Field_Details/Crops_Grown": "maize beans avocado",
  "Field_Details/Field_Size_sqm": "2300",
  "__version__": "veGcULpqP6JNFKRJbbMvMs",
  "meta/instanceID": "uuid:2356cbbe-c1fd-414d-85c8-84f33e92618a",
  "_xform_id_string": "ajXVJpBkTD5tB4Nu9QXpgm",
  "_uid": "2356cbbe-c1fd-414d-85c8-84f33e92618a",
  "_attachments": [],
  "_status": "submitted_via_web",
  "_geolocation": [
    -1.932914,
    30.078211
  ],
  "_submission_time": "2021-12-05T14:15:44",
  "_tags": [],
  "_notes": []
}
```

```

    "_validation_status": {},
    "_submitted_by": null
}

```

With reformatting enabled, and the appropriate masks for multi-select and number formatting (for example, `Crops_*` and `*_sqm` respectively), n8n parses it into:

```

{
  "id": 471987,
  "formhub": {
    "uuid": "189436bb09a54957bfcc798e338b54d6"
  },
  "start": "2021-12-05T16:13:38.527+02:00",
  "end": "2021-12-05T16:15:33.407+02:00",
  "Field_Details": {
    "Field_Name": "Test Fields",
    "Field_Location": {
      "lat": -1.932914,
      "lon": 30.078211
    },
    "Field_Shape": {
      "type": "polygon",
      "coordinates": [
        {
          "lat": -1.932914,
          "lon": 30.078211
        },
        {
          "lat": -1.933011,
          "lon": 30.078085
        },
        {
          "lat": -1.933257,
          "lon": 30.078004
        },
        {
          "lat": -1.933338,
          "lon": 30.078197
        },
        {
          "lat": -1.933107,
          "lon": 30.078299
        },
        {
          "lat": -1.932914,
          "lon": 30.078211
        }
      ]
    },
    "Crops_Grown": [
      "maize",
      "beans",
      "avocado"
    ],
    "Field_Size_sqm": 2300
  },
  "version": "veGcULpqP6JNFKRJbbMvMs",
  "meta": {
    "instanceID": "uuid:2356cbbe-c1fd-414d-85c8-84f33e92618a"
  },

```

```
        "xform_id_string": "ajXVJpBkTD5tB4Nu9QXpgm",
        "uuid": "2356cbbe-c1fd-414d-85c8-84f33e92618a",
        "attachments": [],
        "status": "submitted_via_web",
        "geolocation": {
            "lat": -1.932914,
            "lon": 30.078211
        },
        "submission_time": "2021-12-05T14:15:44",
        "tags": [],
        "notes": [],
        "validation_status": {},
        "submitted_by": null
    }

```

-8<- _snippets/integrations/builtin/app-nodes/operation-not-supported.md"

Lemlist node

Use the Lemlist node to automate work in Lemlist, and integrate Lemlist with other applications. n8n has built-in support for a wide range of Lemlist features, including getting activities, teams and campaigns, as well as creating, updating, and deleting leads.

On this page, you'll find a list of operations the Lemlist node supports and links to more resources.

Operations

- Activity
 - Get Many: Get many activities
- Campaign
 - Get Many: Get many campaigns
 - Get Stats: Get campaign stats
- Enrichment
 - Get: Fetches a previously completed enrichment
 - Enrich Lead: Enrich a lead using an email or LinkedIn URL
 - Enrich Person: Enrich a person using an email or LinkedIn URL
- Lead
 - Create: Create a new lead
 - Delete: Delete an existing lead
 - Get: Get an existing lead
 - Unsubscribe: Unsubscribe an existing lead
- Team
 - Get: Get an existing team
 - Get Credits: Get an existing team's credits
- Unsubscribe
 - Add: Add an email to an unsubscribe list
 - Delete: Delete an email from an unsubscribe list
 - Get Many: Get many unsubscribed emails

Templates and examples

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

Line node

Use the Line node to automate work in Line, and integrate Line with other applications. n8n has built-in support for a wide range of Line features, including sending notifications.

On this page, you'll find a list of operations the Line node supports and links to more resources.

```
-8<- "_snippets/integrations/builtin/app-nodes/ai-tools.md"
```

Operations

- Notification
 - Sends notifications to users or groups

Templates and examples

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

Linear node

Use the Linear node to automate work in Linear, and integrate Linear with other applications. n8n has built-in support for a wide range of Linear features, including creating, updating, deleting, and getting issues.

On this page, you'll find a list of operations the Linear node supports and links to more resources.

Operations

- Comment
 - Add Comment
- Issue
 - Add Link
 - Create
 - Delete
 - Get
 - Get Many
 - Update

Templates and examples

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

LingvaNex node

Use the LingvaNex node to automate work in LingvaNex, and integrate LingvaNex with other applications. n8n has built-in support for translating data with LingvaNex.

On this page, you'll find a list of operations the LingvaNex node supports and links to more resources.

Operations

- Translate data

Templates and examples

LinkedIn node

Use the LinkedIn node to automate work in LinkedIn, and integrate LinkedIn with other applications. n8n supports creating posts.

On this page, you'll find a list of operations the LinkedIn node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Post
 - Create

Parameters

- **Post As:** choose whether to post as a **Person** or **Organization**.
- **Person Name or ID** and **Organization URN:** enter an identifier for the person or organization.
- **Text:** the post contents.
- **Media Category:** use this when including images or article URLs in your post.

Templates and examples

Related resources

Refer to [LinkedIn's API documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

LoneScale node

Use the LoneScale node to automate work in LoneScale and integrate LoneScale with other applications. n8n has built-in support for managing Lists and Items in LoneScale.

On this page, you'll find a list of operations the LoneScale node supports, and links to more resources.

Operations

- List
 - Create
- Item
 - Create

Templates and examples

Related resources

Refer to [LoneScales documentation](#) for more information about the service.

n8n provides a trigger node for LoneScale. You can find the trigger node docs [here](#).

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Magento 2 node

Use the Magento 2 node to automate work in Magento 2, and integrate Magento 2 with other applications. n8n has built-in support for a wide range of Magento 2 features, including creating, updating, deleting, and getting customers, invoices, orders, and projects.

On this page, you'll find a list of operations the Magento 2 node supports and links to more resources.

Operations

- Customer
 - Create a new customer
 - Delete a customer

- Get a customer
- Get all customers
- Update a customer
- Invoice
 - Create an invoice
- Order
 - Cancel an order
 - Get an order
 - Get all orders
 - Ship an order
- Product
 - Create a product
 - Delete a product
 - Get a product
 - Get all products
 - Update a product

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Mailcheck node

Use the Mailcheck node to automate work in Mailcheck, and integrate Mailcheck with other applications. n8n has built-in support for a wide range of Mailcheck features, including checking emails.

On this page, you’ll find a list of operations the Mailcheck node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Email
 - Check

Templates and examples

Mailchimp node

Use the Mailchimp node to automate work in Mailchimp, and integrate Mailchimp with other applications. n8n has built-in support for a wide range of Mailchimp features, including creating, updating, and deleting campaigns, as well as getting list groups.

On this page, you’ll find a list of operations the Mailchimp node supports and links to more resources.

Operations

- Campaign
 - Delete a campaign
 - Get a campaign
 - Get all the campaigns
 - Replicate a campaign
 - Creates a Resend to Non-Openers version of this campaign
 - Send a campaign
- List Group
 - Get all groups
- Member
 - Create a new member on list
 - Delete a member on list
 - Get a member on list
 - Get all members on list
 - Update a new member on list
- Member Tag
 - Add tags from a list member
 - Remove tags from a list member

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

MailerLite node

Use the MailerLite node to automate work in MailerLite, and integrate MailerLite with other applications. n8n has built-in support for a wide range of MailerLite features, including creating, updating, deleting, and getting subscribers.

On this page, you'll find a list of operations the MailerLite node supports and links to more resources.

Operations

- Subscriber
 - Create a new subscriber
 - Get an subscriber
 - Get all subscribers
 - Update an subscriber

Templates and examples

Mailgun node

Use the Mailgun node to automate work in Mailgun, and integrate Mailgun with other applications. n8n has built-in support for sending emails with Mailgun.

On this page, you'll find a list of operations the Mailgun node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Send an email

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Mailjet node

Use the Mailjet node to automate work in Mailjet, and integrate Mailjet with other applications. n8n has built-in support for a wide range of Mailjet features, including sending emails, and SMS.

On this page, you'll find a list of operations the Mailjet node supports and links to more resources.

Operations

- Email
 - Send an email
 - Send an email template
- SMS
 - Send an SMS

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Mandrill node

Use the Mandrill node to automate work in Mandrill, and integrate Mandrill with other applications. n8n supports sending messages based on templates or HTML with Mandrill.

On this page, you'll find a list of operations the Mandrill node supports and links to more resources.

Operations

- Message
 - Send message based on template.
 - Send message based on HTML.

Templates and examples

marketstack node

Use the marketstack node to automate work in marketstack, and integrate marketstack with other applications. n8n has built-in support for a wide range of marketstack features, including getting exchanges, end-of-day data, and tickers.

On this page, you'll find a list of operations the marketstack node supports and links to more resources.

Operations

- End-of-Day Data
 - Get All
- Exchange
 - Get
- Ticker
 - Get

Templates and examples

Matrix node

Use the Matrix node to automate work in Matrix, and integrate Matrix with other applications. n8n has built-in support for a wide range of Matrix features, including getting current user's account information, sending media and messages to a room, and getting room members and messages.

On this page, you'll find a list of operations the Matrix node supports and links to more resources.

Operations

- Account
 - Get current user's account information
- Event
 - Get single event by ID
- Media
 - Send media to a chat room
- Message

- Send a message to a room
- Gets all messages from a room
- Room
 - New chat room with defined settings
 - Invite a user to a room
 - Join a new room
 - Kick a user from a room
 - Leave a room
- Room Member
 - Get all members

Templates and examples

Mattermost node

Use the Mattermost node to automate work in Mattermost, and integrate Mattermost with other applications. n8n has built-in support for a wide range of Mattermost features, including creating, deleting, and getting channels, and users, as well as posting messages, and adding reactions.

On this page, you'll find a list of operations the Mattermost node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Channel
 - Add a user to a channel
 - Create a new channel
 - Soft delete a channel
 - Get a page of members for a channel
 - Restores a soft deleted channel
 - Search for a channel
 - Get statistics for a channel
- Message
 - Soft delete a post, by marking the post as deleted in the database
 - Post a message into a channel
 - Post an ephemeral message into a channel
- Reaction
 - Add a reaction to a post.
 - Remove a reaction from a post
 - Get all the reactions to one or more posts
- User
 - Create a new user
 - Deactivates the user and revokes all its sessions by archiving its user object.
 - Retrieve all users
 - Get a user by email
 - Get a user by ID
 - Invite user to team

Templates and examples

Related resources

Refer to [Mattermost's documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Channel ID field error

If you're not the System Administrator, you might get an error: **there was a problem loading the parameter options from server: “Mattermost error response: You do not have the appropriate permissions.** next to the **Channel ID** field.

Ask your system administrator to grant you the `post:channel` permission.

Find the channel ID

To find the channel ID in Mattermost:

1. Select the channel from the left sidebar.
 2. Select the channel name at the top.
 3. Select **View Info**.
-

Mautic node

Use the Mautic node to automate work in Mautic, and integrate Mautic with other applications. n8n has built-in support for a wide range of Mautic features, including creating, updating, deleting, and getting companies, and contacts, as well as adding and removing campaign contacts.

On this page, you'll find a list of operations the Mautic node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Campaign Contact
 - Add contact to a campaign
 - Remove contact from a campaign
- Company
 - Create a new company
 - Delete a company
 - Get data of a company
 - Get data of all companies
 - Update a company

- Company Contact
 - Add contact to a company
 - Remove a contact from a company
- Contact
 - Create a new contact
 - Delete a contact
 - Edit contact's points
 - Add/remove contacts from/to the don't contact list
 - Get data of a contact
 - Get data of all contacts
 - Send email to contact
 - Update a contact
- Contact Segment
 - Add contact to a segment
 - Remove contact from a segment
- Segment Email
 - Send

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Medium node

Use the Medium node to automate work in Medium, and integrate Medium with other applications. n8n has built-in support for a wide range of Medium features, including creating posts, and getting publications.

On this page, you'll find a list of operations the Medium node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Post
 - Create a post
- Publication
 - Get all publications

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

MessageBird node

Use the MessageBird node to automate work in MessageBird, and integrate MessageBird with other applications. n8n has built-in support for a wide range of MessageBird features, including sending messages, and getting balances.

On this page, you'll find a list of operations the MessageBird node supports and links to more resources.

Operations

- SMS
 - Send text messages (SMS)
- Balance
 - Get the balance

Templates and examples

Metabase node

Use the Metabase node to automate work in Metabase, and integrate Metabase with other applications. n8n has built-in support for a wide range of Metabase features, including adding, and getting alerts, databases, metrics, and questions.

On this page, you'll find a list of operations the Metabase node supports and links to more resources.

Operations

- Alert
 - Get
 - Get All
- Database
 - Add
 - Get All
 - Get Fields
- Metric
 - Get
 - Get All
- Question
 - Get
 - Get All
 - Result Data

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Microsoft Dynamics CRM node

Use the Microsoft Dynamics CRM node to automate work in Microsoft Dynamics CRM, and integrate Microsoft Dynamics CRM with other applications. n8n has built-in support for creating, updating, deleting, and getting Microsoft Dynamics CRM accounts.

On this page, you'll find a list of operations the Microsoft Dynamics CRM node supports and links to more resources.

Operations

- Account
 - Create
 - Delete
 - Get
 - Get All
 - Update

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Microsoft Entra ID node

Use the Microsoft Entra ID node to automate work in Microsoft Entra ID and integrate Microsoft Entra ID with other applications. n8n has built-in support for 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.

On this page, you'll find a list of operations the Microsoft Entra ID node supports, and links to more resources.

Operations

- **Group**
 - **Create:** Create a new group
 - **Delete:** Delete an existing group
 - **Get:** Retrieve data for a specific group
 - **Get Many:** Retrieve a list of groups
 - **Update:** Update a group
- **User**
 - **Create:** Create a new user
 - **Delete:** Delete an existing user
 - **Get:** Retrieve data for a specific user
 - **Get Many:** Retrieve a list of users
 - **Update:** Update a user
 - **Add to Group:** Add user to a group
 - **Remove from Group:** Remove user from a group

Templates and examples

Related resources

Refer to [Microsoft Entra ID's documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Common issues

Here are some common errors and issues with the Microsoft Entra ID node and steps to resolve or troubleshoot them.

Updating the Allow External Senders and Auto Subscribe New Members options fails

You can't update the **Allow External Senders** and **Auto Subscribe New Members** options directly after creating a new group. You must wait after creating a group before you can change the values of these options.

When designing workflows that use multiple Microsoft Entra ID nodes to first create groups and then update these options, add a [Wait](#) node between the two operations. A Wait node configured to pause for at least two seconds allows time for the group to fully initialize. After the wait, the update operation can complete without erroring.

Microsoft Excel 365 node

Use the Microsoft Excel node to automate work in Microsoft Excel, and integrate Microsoft Excel with other applications. n8n has built-in support for a wide range of Microsoft Excel features, including adding and retrieving lists of table data, and workbooks, as well as getting worksheets.

On this page, you'll find a list of operations the Microsoft Excel node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Table
 - Adds rows to the end of the table
 - Retrieve a list of table columns
 - Retrieve a list of table rows
 - Looks for a specific column value and then returns the matching row
- Workbook
 - Adds a new worksheet to the workbook.
 - Get data of all workbooks

- Worksheet
 - Get all worksheets
 - Get worksheet content

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Microsoft Graph Security node

Use the Microsoft Graph Security node to automate work in Microsoft Graph Security, and integrate Microsoft Graph Security with other applications. n8n has built-in support for a wide range of Microsoft Graph Security features, including getting, and updating scores, and profiles.

On this page, you'll find a list of operations the Microsoft Graph Security node supports and links to more resources.

Operations

- Secure Score
 - Get
 - Get All
- Secure Score Control Profile
 - Get
 - Get All
 - Update

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Microsoft OneDrive node

Use the Microsoft OneDrive node to automate work in Microsoft OneDrive, and integrate Microsoft OneDrive with other applications. n8n has built-in support for a wide range of Microsoft OneDrive features, including creating, updating, deleting, and getting files, and folders.

On this page, you'll find a list of operations the Microsoft OneDrive node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- File
 - Copy a file
 - Delete a file
 - Download a file
 - Get a file
 - Rename a file
 - Search a file
 - Share a file
 - Upload a file up to 4MB in size
- Folder
 - Create a folder
 - Delete a folder
 - Get Children (get items inside a folder)
 - Rename a folder
 - Search a folder
 - Share a folder

Templates and examples

Related resources

Refer to [Microsoft's OneDrive API documentation](#) for more information about the service.

Find the folder ID

To perform operations on folders, you need to supply the ID. You can find this:

- In the URL of the folder
- By searching for it using the node. You need to do this if using MS 365 (where OneDrive uses SharePoint behind the scenes):
 1. Select **Resource > Folder**.
 2. Select **Operation > Search**.
 3. In **Query**, enter the folder name.
 4. Select **Execute step**. n8n runs the query and returns data about the folder, including an **id** field containing the folder ID.

Microsoft Outlook node

Use the Microsoft Outlook node to automate work in Microsoft Outlook, and integrate Microsoft Outlook with other applications. n8n has built-in support for a wide range of Microsoft Outlook features, including creating, updating, deleting, and getting folders, messages, and drafts.

On this page, you'll find a list of operations the Microsoft Outlook node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Calendar
 - Create
 - Delete
 - Get
 - Get Many
 - Update
- Contact
 - Create
 - Delete
 - Get
 - Get Many
 - Update
- Draft
 - Create
 - Delete
 - Get
 - Send
 - Update
- Event
 - Create
 - Delete
 - Get
 - Get Many
 - Update
- Folder
 - Create
 - Delete
 - Get
 - Get Many
 - Update
- Folder Message
 - Get Many
- Message
 - Delete
 - Get
 - Get Many
 - Move
 - Reply
 - Send
 - Send and Wait for Response
 - Update
- Message Attachment
 - Add
 - Download
 - Get
 - Get Many

-8<- “_snippets/integrations/builtin/send-and-wait-operation.md”

Templates and examples

Related resources

Refer to [Outlook's API documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Microsoft SharePoint node

Use the Microsoft SharePoint node to automate work in Microsoft SharePoint and integrate Microsoft SharePoint with other applications. n8n has built-in support for a wide range of Microsoft SharePoint features, which includes downloading, uploading, and updating files, managing items in a list, and getting lists and list items.

On this page, you'll find a list of operations the Microsoft SharePoint node supports, and links to more resources.

Operations

- **File:**
 - Download: Download a file.
 - Update: Update a file.
 - Upload: Upload an existing file.
- **Item:**
 - Create: Create an item in an existing list.
 - Create or Update: Create a new item, or update the current one if it already exists (upsert).
 - Delete: Delete an item from a list.
 - Get: Retrieve an item from a list.
 - Get Many: Get specific items in a list or list many items.
 - Update: Update an item in an existing list.
- **List:**
 - Get: Retrieve details of a single list.
 - Get Many: Retrieve a list of lists.

Templates and examples

Related resources

Refer to [Microsoft's SharePoint documentation](#) for more information about the service.

Microsoft SQL node

Use the Microsoft SQL node to automate work in Microsoft SQL, and integrate Microsoft SQL with other applications. n8n has built-in support for a wide range of Microsoft SQL features, including executing SQL queries, and inserting rows into the database.

On this page, you'll find a list of operations the Microsoft SQL node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Execute an SQL query
- Insert rows in database
- Update rows in database
- Delete rows in database

Templates and examples

Microsoft Teams node

Use the Microsoft Teams node to automate work in Microsoft Teams, and integrate Microsoft Teams with other applications. n8n has built-in support for a wide range of Microsoft Teams features, including creating and deleting, channels, messages, and tasks.

On this page, you'll find a list of operations the Microsoft Teams node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Channel
 - Create
 - Delete
 - Get
 - Get Many
 - Update
- Channel Message
 - Create
 - Get Many
- Chat Message
 - Create
 - Get
 - Get Many
 - Send and Wait for Response
- Task
 - Create
 - Delete
 - Get
 - Get Many
 - Update

-8<- “_snippets/integrations/builtin/send-and-wait-operation.md”

Templates and examples

Related resources

Refer to [Microsoft Teams' API documentation](#) for more information about the service.

```
-8<- "snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

Microsoft To Do node

Use the Microsoft To Do node to automate work in Microsoft To Do, and integrate Microsoft To Do with other applications. n8n has built-in support for a wide range of Microsoft To Do features, including creating, updating, deleting, and getting linked resources, lists, and tasks.

On this page, you'll find a list of operations the Microsoft To Do node supports and links to more resources.

```
-8<- "snippets/integrations/builtin/app-nodes/ai-tools.md"
```

Operations

- Linked Resource
 - Create
 - Delete
 - Get
 - Get All
 - Update
- List
 - Create
 - Delete
 - Get
 - Get All
 - Update
- Task
 - Create
 - Delete
 - Get
 - Get All
 - Update

Templates and examples

```
-8<- "snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

Mindee node

Use the Mindee node to automate work in Mindee, and integrate Mindee with other applications. n8n has built-in support for a wide range of Mindee features, including predicting invoices.

On this page, you'll find a list of operations the Mindee node supports and links to more resources.

Operations

- **Invoice**
 - Predict
- **Receipt**
 - Predict

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

MISP node

Use the MISP node to automate work in MISP, and integrate MISP with other applications. n8n has built-in support for a wide range of MISP features, including creating, updating, deleting and getting events, feeds, and organizations.

On this page, you’ll find a list of operations the MISP node supports and links to more resources.

Operations

- Attribute
 - Create
 - Delete
 - Get
 - Get All
 - Search
 - Update
- Event
 - Create
 - Delete
 - Get
 - Get All
 - Publish
 - Search
 - Unpublish
 - Update
- Event Tag
 - Add
 - Remove
- Feed
 - Create
 - Disable
 - Enable
 - Get
 - Get All
 - Update
- Galaxy
 - Delete
 - Get
 - Get All

- Noticelist
 - Get
 - Get All
- Object
 - Search
- Organisation
 - Create
 - Delete
 - Get
 - Get All
 - Update
- Tag
 - Create
 - Delete
 - Get All
 - Update
- User
 - Create
 - Delete
 - Get
 - Get All
 - Update
- Warninglist
 - Get
 - Get All

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Mistral AI node

Use the Mistral AI node to automate work in Mistral AI and integrate Mistral AI with other applications. n8n has built-in support for extracting text with various models, file types, and input methods.

On this page, you’ll find a list of operations the Mistral AI node supports, and links to more resources.

Node parameters

- **Resource:** The resource that Mistral AI should operate on. The current implementation supports the “Document” resource.
- **Operation:** The operation to perform:
 - **Extract Text:** Extracts text from a document or image using optical character recognition (OCR).
- **Model:** The model to use for the given operation. The current version requires the `mistral-ocr-latest` model.
- **Document Type:** The document format to process. Can be “Document” or “Image”.
- **Input Type:** How to input the document:
 - **Binary Data:** Pass the document to this node as a binary field.
 - **URL:** Fetch the document from a given URL.

- **Input Binary Field:** When using the “Binary Data” input type, defines the name of the input binary field containing the file.
- **URL:** When using the “URL” input type, the URL of the document or image to process.

Node options

- **Enable Batch Processing:** Whether to process multiple documents in the same API call. This may reduce your costs by bundling requests.
- **Batch Size:** When using “Enable Batch Processing”, sets the maximum number of documents to process per batch.
- **Delete Files After Processing:** When using “Enable Batch Processing”, whether to delete the files from Mistral Cloud after processing.

Templates and examples

Related resources

Refer to [Mistral AI’s documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Mocean node

Use the Mocean node to automate work in Mocean, and integrate Mocean with other applications. n8n has built-in support for a wide range of Mocean features, including sending SMS, and voice messages.

On this page, you’ll find a list of operations the Mocean node supports and links to more resources.

Operations

- SMS
 - Send SMS/Voice message
- Voice
 - Send SMS/Voice message

Templates and examples

monday.com node

Use the monday.com node to automate work in monday.com, and integrate monday.com with other applications. n8n has built-in support for a wide range of monday.com features, including creating a new board, and adding, deleting, and getting items on the board.

On this page, you'll find a list of operations the monday.com node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Board
 - Archive a board
 - Create a new board
 - Get a board
 - Get all boards
- Board Column
 - Create a new column
 - Get all columns
- Board Group
 - Delete a group in a board
 - Create a group in a board
 - Get list of groups in a board
- Board Item
 - Add an update to an item.
 - Change a column value for a board item
 - Change multiple column values for a board item
 - Create an item in a board's group
 - Delete an item
 - Get an item
 - Get all items
 - Get items by column value
 - Move item to group

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

MongoDB node

Use the MongoDB node to automate work in MongoDB, and integrate MongoDB with other applications. n8n has built-in support for a wide range of MongoDB features, including aggregating, updating, finding, deleting, and getting documents as well as creating, updating, listing, and dropping search indexes. All operations in this Node make use of the [MongoDB Node driver](#).

On this page, you'll find a list of operations the MongoDB node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Document
 - Aggregate documents
 - Delete documents
 - Find documents
 - Find and replace documents
 - Find and update documents
 - Insert documents
 - Update documents
- Search Index
 - Create search indexes
 - Drop search indexes
 - List search indexes
 - Update search indexes

Templates and examples

Monica CRM node

Use the Monica CRM node to automate work in Monica CRM, and integrate Monica CRM with other applications. n8n has built-in support for a wide range of Monica CRM features, including creating, updating, deleting, and getting activities, calls, contracts, messages, tasks, and notes.

On this page, you'll find a list of operations the Monica CRM node supports and links to more resources.

Operations

- Activity
 - Create an activity
 - Delete an activity
 - Retrieve an activity
 - Retrieve all activities
 - Update an activity
- Call
 - Create a call
 - Delete a call
 - Retrieve a call
 - Retrieve all calls
 - Update a call
- Contact
 - Create a contact
 - Delete a contact
 - Retrieve a contact
 - Retrieve all contacts
 - Update a contact
- Contact Field
 - Create a contact field
 - Delete a contact field
 - Retrieve a contact field
 - Update a contact field

- Contact Tag
 - Add
 - Remove
- Conversation
 - Create a conversation
 - Delete a conversation
 - Retrieve a conversation
 - Update a conversation
- Conversation Message
 - Add a message to a conversation
 - Update a message in a conversation
- Journal Entry
 - Create a journal entry
 - Delete a journal entry
 - Retrieve a journal entry
 - Retrieve all journal entries
 - Update a journal entry
- Note
 - Create a note
 - Delete a note
 - Retrieve a note
 - Retrieve all notes
 - Update a note
- Reminder
 - Create a reminder
 - Delete a reminder
 - Retrieve a reminder
 - Retrieve all reminders
 - Update a reminder
- Tag
 - Create a tag
 - Delete a tag
 - Retrieve a tag
 - Retrieve all tags
 - Update a tag
- Task
 - Create a task
 - Delete a task
 - Retrieve a task
 - Retrieve all tasks
 - Update a task

Templates and examples

MQTT node

Use the MQTT node to automate work in MQTT, and integrate MQTT with other applications. n8n supports transporting messages with MQTT.

On this page, you'll find a list of operations the MQTT node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

Use the MQTT node to send a message. You can set the message topic, and choose whether to send the node input data as part of the message.

Templates and examples

Related resources

n8n provides a trigger node for MQTT. You can find the trigger node docs [here](#).

Refer to [MQTT's documentation](#) for more information about the service.

MSG91 node

Use the MSG91 node to automate work in MSG91, and integrate MSG91 with other applications. n8n supports sending SMS with MSG91.

On this page, you'll find a list of operations the MSG91 node supports and links to more resources.

Operations

- SMS
 - Send SMS

Templates and examples

Find your Sender ID

1. Log in to your MSG91 dashboard.
 2. Select **Sender Id** in the left panel.
 3. If you don't already have one, select **Add Sender Id +**, fill in the details, and select **Save Sender Id**.
-

MySQL node

Use the MySQL node to automate work in MySQL, and integrate MySQL with other applications. n8n has built-in support for a wide range of MySQL features, including executing an SQL query, as well as inserting, and updating rows in a database.

On this page, you'll find a list of operations the MySQL node supports and links to more resources.

```
-8<- "_snippets/integrations/builtin/app-nodes/ai-tools.md"
```

Operations

- Delete
- Execute SQL
- Insert
- Insert or Update
- Select
- Update

Templates and examples

Related resources

Refer to [MySQL's Connectors and APIs documentation](#) for more information about the service.

Refer to MySQL's [SELECT statement documentation](#) for more information on writing SQL queries.

Use query parameters

When creating a query to run on a MySQL database, you can use the **Query Parameters** field in the **Options** section to load data into the query. n8n sanitizes data in query parameters, which prevents SQL injection.

For example, you want to find a person by their email address. Given the following input data:

```
[  
  {  
    "email": "alex@example.com",  
    "name": "Alex",  
    "age": 21  
  },  
  {  
    "email": "jamie@example.com",  
    "name": "Jamie",  
    "age": 33  
  }  
]
```

You can write a query like:

```
SELECT * FROM $1:name WHERE email = $2;
```

Then in **Query Parameters**, provide the field values to use. You can provide fixed values or expressions. For this example, use expressions so the node can pull the email address from each input item in turn:

```
// users is an example table name  
users, {{ $json.email }}
```

Common issues

For common errors or issues and suggested resolution steps, refer to [Common issues](#).

MySQL node common issues

Here are some common errors and issues with the [MySQL node](#) and steps to resolve or troubleshoot them.

Update rows by composite key

The MySQL node's **Update** operation lets you update rows in a table by providing a **Column to Match On** and a value. This works for tables where single column values can uniquely identify individual rows.

You can't use this pattern for tables that use [composite keys](#), where you need multiple columns to uniquely identify a row. An example of this is MySQL's [user table](#) in the `mysql` database, where you need both the `user` and `host` columns to uniquely identify rows.

To update tables with composite keys, write the query manually with the **Execute SQL** operation instead. There, you can match on multiple values, like in this example which matches on both `customer_id` and `product_id`:

```
UPDATE orders SET quantity = 3 WHERE customer_id = 538 AND product_id = 800;
```

Can't connect to a local MySQL server when using Docker

When you run either n8n or MySQL in Docker, you need to configure the network so that n8n can connect to MySQL.

The solution depends on how you're hosting the two components.

If only MySQL is in Docker

If only MySQL is running in Docker, configure MySQL to listen on all interfaces by binding to `0.0.0.0` inside of the container (the official images are already configured this way).

When running the container, [publish the port](#) with the `-p` flag. By default, MySQL runs on port 3306, so your Docker command should look like this:

```
docker run -p 3306:3306 --name my-mysql -d mysql:latest
```

When configuring [MySQL credentials](#), the `localhost` address should work without a problem (set the **Host** to `localhost`).

If only n8n is in Docker

If only n8n is running in Docker, configure MySQL to listen on all interfaces by binding to `0.0.0.0` on the host.

If you are running n8n in Docker on **Linux**, use the `--add-host` flag to map `host.docker.internal` to `host-gateway` when you start the container. For example:

```
docker run -it --rm --add-host host.docker.internal:host-gateway --  
name n8n -p 5678:5678 -v n8n_data:/home/node/.n8n  
docker.n8n.io/n8nio/n8n
```

If you are using Docker Desktop, this is automatically configured for you.

When configuring [MySQL credentials](#), use `host.docker.internal` as the **Host** address instead of `localhost`.

If MySQL and n8n are running in separate Docker containers

If both n8n and MySQL are running in Docker in separate containers, you can use Docker networking to connect them.

Configure MySQL to listen on all interfaces by binding to `0.0.0.0` inside of the container (the official images are already configured this way). Add both the MySQL and n8n containers to the same [user-defined bridge network](#).

When configuring [MySQL credentials](#), use the MySQL container's name as the host address instead of `localhost`. For example, if you call the MySQL container `my-mysql`, you would set the **Host** to `my-mysql`.

If MySQL and n8n are running in the same Docker container

If MySQL and n8n are running in the same Docker container, the `localhost` address doesn't need any special configuration. You can configure MySQL to listen on `localhost` and configure the **Host** in the [MySQL credentials](#) in n8n to use `localhost`.

Decimal numbers returned as strings

By default, the MySQL node returns [DECIMAL values](#) as strings. This is done intentionally to avoid loss of precision that can occur due to limitation with the way JavaScript represents numbers. You can learn more about the decision in the documentation for the [MySQL library](#) that n8n uses.

To output decimal values as numbers instead of strings and ignore the risks in loss of precision, enable the **Output Decimals as Numbers** option. This will output the values as numbers instead of strings.

As an alternative, you can manually convert from the string to a decimal using the [toFloat\(\) function](#) with [toFixed\(\)](#) or with the [Edit Fields \(Set\) node](#) after the MySQL node. Be aware that you may still need to account for a potential loss of precision.

Customer Datastore (n8n Training) node

Use this node only for the n8n new user onboarding tutorial. It provides dummy data for testing purposes and has no further functionality.

Customer Messenger (n8n Training) node

Use this node only for the n8n new user onboarding tutorial. It provides no further functionality.

NASA node

Use the NASA node to automate work in NASA, and integrate NASA with other applications. n8n has built-in support for a wide range of NASA features, including retrieving imagery and data.

On this page, you'll find a list of operations the NASA node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Astronomy Picture of the Day
 - Get the Astronomy Picture of the Day
- Asteroid Neo-Feed
 - Retrieve a list of asteroids based on their closest approach date to Earth
- Asteroid Neo-Lookup
 - Look up an asteroid based on its NASA SPK-ID
- Asteroid Neo-Browse
 - Browse the overall asteroid dataset
- DONKI Coronal Mass Ejection
 - Retrieve DONKI coronal mass ejection data
- DONKI Interplanetary Shock
 - Retrieve DONKI interplanetary shock data
- DONKI Solar Flare
 - Retrieve DONKI solar flare data
- DONKI Solar Energetic Particle
 - Retrieve DONKI solar energetic particle data
- DONKI Magnetopause Crossing
 - Retrieve data on DONKI magnetopause crossings
- DONKI Radiation Belt Enhancement
 - Retrieve DONKI radiation belt enhancement data
- DONKI High Speed Stream
 - Retrieve DONKI high speed stream data

- DONKI WSA+EnlilSimulation
 - Retrieve DONKI WSA+EnlilSimulation data
- DONKI Notifications
 - Retrieve DONKI notifications data
- Earth Imagery
 - Retrieve Earth imagery
- Earth Assets
 - Retrieve Earth assets

Templates and examples

Netlify node

Use the Netlify node to automate work in Netlify, and integrate Netlify with other applications. n8n has built-in support for a wide range of Netlify features, including getting and cancelling deployments, as well as deleting, and getting sites.

On this page, you'll find a list of operations the Netlify node supports and links to more resources.

Operations

- Deploy
 - Cancel a deployment
 - Create a new deployment
 - Get a deployment
 - Get all deployments
- Site
 - Delete a site
 - Get a site
 - Returns all sites

Templates and examples

Netscaler ADC node

Use the Netscaler ADC node to automate work in Netscaler ADC, and integrate Netscaler ADC with other applications. n8n has built-in support for a wide range of Netscaler ADC features, including creating and installing certificates and files.

On this page, you'll find a list of operations the Netscaler ADC node supports and links to more resources.

Operations

- Certificate
 - Create

- Install
- File
 - Delete
 - Download
 - Upload

Templates and examples

Related resources

Refer to [Netscaler ADC's documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Nextcloud node

Use the Nextcloud node to automate work in Nextcloud, and integrate Nextcloud with other applications. n8n has built-in support for a wide range of Nextcloud features, including creating, updating, deleting, and getting files, and folders as well as retrieving, and inviting users.

On this page, you'll find a list of operations the Nextcloud node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- File
 - Copy a file
 - Delete a file
 - Download a file
 - Move a file
 - Share a file
 - Upload a file
- Folder
 - Copy a folder
 - Create a folder
 - Delete a folder
 - Return the contents of a given folder
 - Move a folder
 - Share a folder
- User
 - Invite a user to a Nextcloud organization
 - Delete a user.
 - Retrieve information about a single user.
 - Retrieve a list of users.
 - Edit attributes related to a user.

Templates and examples

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

NocoDB node

Use the NocoDB node to automate work in NocoDB, and integrate NocoDB with other applications. n8n has built-in support for a wide range of NocoDB features, including creating, updating, deleting, and retrieving rows.

On this page, you'll find a list of operations the NocoDB node supports and links to more resources.

```
-8<- "_snippets/integrations/builtin/app-nodes/ai-tools.md"
```

Operations

- Row
 - Create
 - Delete
 - Get
 - Get Many
 - Update a row

Templates and examples

Relates resources

Refer to [NocoDB's documentation](#) for more information about the service.

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

Notion node

Use the Notion node to automate work in Notion, and integrate Notion with other applications. n8n has built-in support for a wide range of Notion features, including getting and searching databases, creating pages, and getting users.

On this page, you'll find a list of operations the Notion node supports and links to more resources.

```
-8<- "_snippets/integrations/builtin/app-nodes/ai-tools.md"
```

Operations

- Block
 - Append After

- Get Child Blocks
- Database
 - Get
 - Get Many
 - Search
- Database Page
 - Create
 - Get
 - Get Many
 - Update
- Page
 - Archive
 - Create
 - Search
- User
 - Get
 - Get Many

Templates and examples

Related resources

n8n provides an app node for Notion. You can find the trigger node docs [here](#).

Refer to [Notion's documentation](#) for details about their API.

-8<- “ snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Common issues

For common errors or issues and suggested resolution steps, refer to [Common issues](#).

Notion node common issues

Here are some common errors and issues with the [Notion node](#) and steps to resolve or troubleshoot them.

Relation property not displaying

The Notion node only supports displaying the data relation property for [two-way relations](#). When you connect two Notion databases with a two-way relationship, you can select or filter by the relation property when working with the Notion node's **Database Page** resource.

To enable two-way relations, edit the relation property in Notion and enable the **Show on [name of related database]** option to create a reverse relation. Select a name to use for the relation in the new context. The relation is now accessible in n8n when filtering or selecting.

If you need to work with Notion databases with one-way relationship, you can use the [HTTP Request](#) with your existing Notion credentials. For example, to update a one-way relationship, you can send a PATCH request to the following URL:

```
https://api.notion.com/v1/pages/<page_id>
```

Enable **Send Body**, set the **Body Content Type** to **JSON**, and set **Specify Body** to **Using JSON**. Afterward, you can enter a JSON object like the following into the **JSON** field:

```
{
  "properties": {
    "Account": {
      "relation": [
        {
          "id": "<your_relation_ID>"
        }
      ]
    }
  }
}
```

Create toggle heading

The Notion node allows you to create headings and toggles when adding blocks to **Page**, **Database Page**, or **Block** resources. Creating toggleable headings isn't yet supported by the Notion node itself.

You can work around this by creating a regular heading and then modifying it to enable the [is_toggleable](#) property:

1. Add a heading with Notion node.
2. Select the resource you want to add a heading to:
 - To add a new page with a heading, select the **Page** or **Database Page** resources with the **Create** operation.
 - To add a heading to an existing page, select the **Block** resource with the **Append After** operation.
3. Select **Add Block** and set the **Type Name or ID** to either **Heading 1**, **Heading 2**, or **Heading 3**.
4. Add an [HTTP Request](#) node connected to the Notion node and select the **GET** method.
5. Set the **URL** to https://api.notion.com/v1/blocks/<block_ID>. For example, if you added the heading to an existing page, you could use the following URL: [https://api.notion.com/v1/blocks/{{ \\$json.results\[0\].id }}](https://api.notion.com/v1/blocks/{{ $json.results[0].id }}). If you created a new page instead of appending a block, you may need to discover the block ID by querying the page contents first.
6. Select **Predefined Credential Type** and connect your existing Notion credentials.
7. Add an [Edit Fields \(Set\)](#) node after the HTTP Request node.
8. Add **heading_1.is_toggleable** as a new **Boolean** field set to **true**. Swap **heading_1** for a different heading number as necessary.
9. Add a second HTTP Request node after the Edit Fields (Set) node.
10. Set the **Method** to **PATCH** and use [https://api.notion.com/v1/blocks/{{ \\$json.id }}](https://api.notion.com/v1/blocks/{{ $json.id }}) as the **URL** value.
11. Select **Predefined Credential Type** and connect your existing Notion credentials.
12. Enable **Send Body** and set a parameter.

13. Set the parameter **Name** to `heading_1` (substitute `heading_1` for the heading level you are using).
14. Set the parameter **Value** to `{{ $json.heading_1 }}` (substitute `heading_1` for the heading level you are using).

The above sequence will create a regular heading block. It will query the newly created header, add the `is_toggleable` property, and update the heading block.

Handle null and empty values

You may receive a validation error when working with the Notion node if you submit fields with empty or null values. This can occur any time you populate fields from previous nodes when that data is missing.

To work around this, check for the existence of the field data before sending it to Notion or use a default value.

To check for the data before executing the Notion node, use an If node to check whether the field is unset. This allows you to use the Edit Fields (Set) node to conditionally remove the field when it doesn't have a valid value.

As an alternative, you can set a default value if the incoming data doesn't provide one.

npm node

Use the npm node to automate work in npm, and integrate npm with other applications.

On this page, you'll find a list of operations the npm node supports and links to more resources.

Operations

- Package
 - Get Package Metadata
 - Get Package Versions
 - Search for Packages
- Distribution Tag
 - Get All Tags
 - Update a Tag

Templates and examples

Related resources

Refer to [npm's documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Odoo node

Use the Odoo node to automate work in Odoo, and integrate Odoo with other applications. n8n has built-in support for a wide range of Odoo features, including creating, updating, deleting, and getting contracts, resources, and opportunities.

On this page, you'll find a list of operations the Odoo node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Contact
 - Create a new contact
 - Delete a contact
 - Get a contact
 - Get all contacts
 - Update a contact
- Custom Resource
 - Create a new item
 - Delete an item
 - Get an item
 - Get all items
 - Update an item
- Note
 - Create a new note
 - Delete a note
 - Get a note
 - Get all notes
 - Update a note
- Opportunity
 - Create a new opportunity
 - Delete an opportunity
 - Get an opportunity
 - Get all opportunities
 - Update an opportunity

Templates and examples

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.

On this page, you'll find a list of operations the Okta node supports, and links to more resources.

Operations

- User

- Create a new user
- Delete an existing user
- Get details of a user
- Get many users
- Update an existing user

Templates and examples

Related resources

Refer to [Okta's documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

One Simple API node

Use the One Simple API node to automate work in One Simple API, and integrate One Simple API with other applications. n8n has built-in support for a wide range of One Simple API features, including getting profiles, retrieving information, and generating utilities.

On this page, you'll find a list of operations the One Simple API node supports and links to more resources.

Operations

- Information
 - Convert a value between currencies
 - Retrieve image metadata from a URL
- Social Profile
 - Get details about an Instagram profile
 - Get details about a Spotify Artist
- Utility
 - Expand a shortened url
 - Generate a QR Code
 - Validate an email address
- Website
 - Generate a PDF from a webpage
 - Get SEO information from website
 - Create a screenshot from a webpage

Templates and examples

Related resources

Refer to [One Simple API's documentation](#) for more information about the service.

Onfleet node

Use the Onfleet node to automate work in Onfleet, and integrate Onfleet with other applications. n8n has built-in support for a wide range of Onfleet features, including creating and deleting tasks in Onfleet as well as retrieving organizations' details.

On this page, you'll find a list of operations the Onfleet node supports and links to more resources.

Operations

- Admin
 - Create a new Onfleet admin
 - Delete an Onfleet admin
 - Get all Onfleet admins
 - Update an Onfleet admin
- Container
 - Add task at index (or append)
 - Get container information
 - Fully replace a container's tasks
- Destination
 - Create a new destination
 - Get a specific destination
- Hub
 - Create a new Onfleet hub
 - Get all Onfleet hubs
 - Update an Onfleet hub
- Organization
 - Retrieve your own organization's details
 - Retrieve the details of an organization with which you are connected
- Recipient
 - Create a new Onfleet recipient
 - Get a specific Onfleet recipient
 - Update an Onfleet recipient
- Task
 - Create a new Onfleet task
 - Clone an Onfleet task
 - Force-complete a started Onfleet task
 - Delete an Onfleet task
 - Get all Onfleet tasks
 - Get a specific Onfleet task
 - Update an Onfleet task
- Team
 - Automatically dispatch tasks assigned to a team to on-duty drivers
 - Create a new Onfleet team
 - Delete an Onfleet team
 - Get a specific Onfleet team
 - Get all Onfleet teams
 - Get estimated times for upcoming tasks for a team, returns a selected driver
 - Update an Onfleet team
- Worker
 - Create a new Onfleet worker
 - Delete an Onfleet worker
 - Get a specific Onfleet worker

- Get all Onfleet workers
- Get a specific Onfleet worker schedule
- Update an Onfleet worker

Templates and examples

OpenAI node

Use the OpenAI node to automate work in OpenAI and integrate OpenAI with other applications. n8n has built-in support for a wide range of OpenAI features, including creating images and assistants, as well as chatting with models.

On this page, you'll find a list of operations the OpenAI node supports and links to more resources.

Operations

- **Text**
 - [Generate a Chat Completion](#)
 - [Generate a Model Response](#)
 - [Classify Text for Violations](#)
- **Image**
 - [Analyze Image](#)
 - [Generate an Image](#)
 - [Edit an Image](#)
- **Audio**
 - [Generate Audio](#)
 - [Transcribe a Recording](#)
 - [Translate a Recording](#)
- **File**
 - [Delete a File](#)
 - [List Files](#)
 - [Upload a File](#)
- **Video**
 - [Generate a Video](#)
- **Conversation**
 - [Create a Conversation](#)
 - [Get a Conversation](#)
 - [Update a Conversation](#)
 - [Remove a Conversation](#)

Templates and examples

Related resources

Refer to [OpenAI's documentation](#) for more information about the service.

Refer to [OpenAI's assistants documentation](#) for more information about how assistants work.

For help dealing with rate limits, refer to [Handling rate limits](#).

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Using tools with OpenAI assistants

Some operations allow you to connect tools. **Tools** act like addons that your AI can use to access extra context or resources.

Select the **Tools** connector to browse the available tools and add them.

Once you add a tool connection, the OpenAI node becomes a [root node](#), allowing it to form a [cluster node](#) with the tools [sub-nodes](#). See [Node types](#) for more information on cluster nodes and root nodes.

Operations that support tool connectors

- **Text**
 - [Generate a Chat Completion](#)
 - [Generate a Model Response](#)

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

OpenAI Assistant operations

Use this operation to create, delete, list, message, or update an assistant in OpenAI. Refer to [OpenAI](#) for more information on the OpenAI node itself.

Create an Assistant

Use this operation to create a new assistant.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **Assistant**.
- **Operation:** Select **Create an Assistant**.
- **Model:** Select the model that the assistant will use. If you’re not sure which model to use, try gpt-4o if you need high intelligence or gpt-4o-mini if you need the fastest speed and lowest cost. Refer to [Models overview | OpenAI Platform](#) for more information.
- **Name:** Enter the name of the assistant. The maximum length is 256 characters.

- **Description:** Enter the description of the assistant. The maximum length is 512 characters.

A virtual assistant that helps users with daily tasks, including setting reminders, answering general questions, and providing quick information.

- **Instructions:** Enter the system instructions that the assistant uses. The maximum length is 32,768 characters. Use this to specify the persona used by the model in its replies.

Always respond in a friendly and engaging manner. When a user asks a question, provide a concise answer first, followed by a brief explanation or additional context if necessary. If the question is open-ended, offer a suggestion or ask a clarifying question to guide the conversation. Keep the tone positive and supportive, and avoid technical jargon unless specifically requested by the user.

- **Code Interpreter:** Turn on to enable the code interpreter for the assistant, where it can write and execute code in a sandbox environment. Enable this tool for tasks that require computations, data analysis, or any logic-based processing.

- **Knowledge Retrieval:** Turn on to enable knowledge retrieval for the assistant, allowing it to access external sources or a connected knowledge base. Refer to [File Search | OpenAI Platform](#) for more information.

- **Files:** Select a file to upload for your external knowledge source. Use **Upload a File** operation to add more files.

Options

- **Output Randomness (Temperature):** Adjust the randomness of the response. The range is between 0.0 (deterministic) and 1.0 (maximum randomness). We recommend altering this or **Output Randomness (Top P)** but not both. Start with a medium temperature (around 0.7) and adjust based on the outputs you observe. If the responses are too repetitive or rigid, increase the temperature. If they're too chaotic or off-track, decrease it. Defaults to 1.0.
- **Output Randomness (Top P):** Adjust the Top P setting to control the diversity of the assistant's responses. For example, 0.5 means half of all likelihood-weighted options are considered. We recommend altering this or **Output Randomness (Temperature)** but not both. Defaults to 1.0.
- **Fail if Assistant Already Exists:** If enabled, the operation will fail if an assistant with the same name already exists.

Refer to [Create assistant | OpenAI](#) documentation for more information.

Delete an Assistant

Use this operation to delete an existing assistant from your account.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **Assistant**.
- **Operation:** Select **Delete an Assistant**.
- **Assistant:** Select the assistant you want to delete **From list** or **By ID**.

Refer to [Delete assistant | OpenAI](#) documentation for more information.

List Assistants

Use this operation to retrieve a list of assistants in your organization.

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **Assistant**.
- **Operation:** Select **List Assistants**.

Options

- **Simplify Output:** Turn on to return a simplified version of the response instead of the raw data. This option is enabled by default.

Refer to [List assistants | OpenAI](#) documentation for more information.

Message an Assistant

Use this operation to send a message to an assistant and receive a response.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **Assistant**.
- **Operation:** Select **Message an Assistant**.
- **Assistant:** Select the assistant you want to message.
- **Prompt:** Enter the text prompt or message that you want to send to the assistant.
 - **Connected Chat Trigger Node:** Automatically use the input from a previous node's chatInput field.
 - **Define Below:** Manually define the prompt by entering static text or using an expression to reference data from previous nodes.

Options

- **Base URL:** Enter the base URL that the assistant should use for making API requests. This option is useful for directing the assistant to use endpoints provided by other LLM providers that offer an OpenAI-compatible API.
- **Max Retries:** Specify the number of times the assistant should retry an operation in case of failure.
- **Timeout:** Set the maximum amount of time in milliseconds, that the assistant should wait for a response before timing out. Use this

option to prevent long waits during operations.

- **Preserve Original Tools:** Turn off to remove the original tools associated with the assistant. Use this if you want to temporarily remove tools for this specific operation.

Refer to [Assistants | OpenAI](#) documentation for more information.

Update an Assistant

Use this operation to update the details of an existing assistant.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **Assistant**.
- **Operation:** Select **Update an Assistant**.
- **Assistant:** Select the assistant you want to update.

Options

- **Code Interpreter:** Turn on to enable the code interpreter for the assistant, where it can write and execute code in a sandbox environment. Enable this tool for tasks that require computations, data analysis, or any logic-based processing.

- **Description:** Enter the description of the assistant. The maximum length is 512 characters.

A virtual assistant that helps users with daily tasks, including setting reminders, answering general questions, and providing quick information.

- **Instructions:** Enter the system instructions that the assistant uses. The maximum length is 32,768 characters. Use this to specify the persona used by the model in its replies.

Always respond in a friendly and engaging manner. When a user asks a question, provide a concise answer first, followed by a brief explanation or additional context if necessary. If the question is open-ended, offer a suggestion or ask a clarifying question to guide the conversation. Keep the tone positive and supportive, and avoid technical jargon unless specifically requested by the user.

- **Knowledge Retrieval:** Turn on to enable knowledge retrieval for the assistant, allowing it to access external sources or a connected knowledge base. Refer to [File Search | OpenAI Platform](#) for more information.

- **Files:** Select a file to upload for your external knowledge source. Use **Upload a File** operation to add more files. Note that this only updates the Code Interpreter tool, not the File Search tool.

- **Model:** Select the model that the assistant will use. If you're not sure which model to use, try gpt-4o if you need high intelligence or gpt-4o-mini if you need the fastest speed and lowest cost. Refer to [Models overview | OpenAI Platform](#) for more information.

- **Name:** Enter the name of the assistant. The maximum length is 256 characters.
- **Remove All Custom Tools (Functions):** Turn on to remove all custom tools (functions) from the assistant.
- **Output Randomness (Temperature):** Adjust the randomness of the response. The range is between 0.0 (deterministic) and 1.0 (maximum randomness). We recommend altering this or **Output Randomness (Top P)** but not both. Start with a medium temperature (around 0.7) and adjust based on the outputs you observe. If the responses are too repetitive or rigid, increase the temperature. If they're too chaotic or off-track, decrease it. Defaults to 1.0.
- **Output Randomness (Top P):** Adjust the Top P setting to control the diversity of the assistant's responses. For example, 0.5 means half of all likelihood-weighted options are considered. We recommend altering this or **Output Randomness (Temperature)** but not both. Defaults to 1.0.

Refer to [Modify assistant | OpenAI](#) documentation for more information.

Common issues

For common errors or issues and suggested resolution steps, refer to [Common Issues](#).

OpenAI Audio operations

Use this operation to generate an audio, or transcribe or translate a recording in OpenAI. Refer to [OpenAI](#) for more information on the OpenAI node itself.

Generate Audio

Use this operation to create audio from a text prompt.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **Audio**.
- **Operation:** Select **Generate Audio**.
- **Model:** Select the model you want to use to generate the audio. Refer to [TTS | OpenAI](#) for more information.
 - **TTS-1:** Use this to optimize for speed.
 - **TTS-1-HD:** Use this to optimize for quality.
- **Text Input:** Enter the text to generate the audio for. The maximum length is 4096 characters.
- **Voice:** Select a voice to use when generating the audio. Listen to the previews of the voices in [Text to speech guide | OpenAI](#).

Options

- **Response Format:** Select the format for the audio response. Choose from **MP3** (default), **OPUS**, **AAC**, **FLAC**, **WAV**, and **PCM**.
- **Audio Speed:** Enter the speed for the generated audio from a value from 0.25 to 4.0. Defaults to 1.
- **Put Output in Field:** Defaults to data. Enter the name of the output field to put the binary file data in.

Refer to [Create speech | OpenAI](#) documentation for more information.

Transcribe a Recording

Use this operation to transcribe audio into text. OpenAI API limits the size of the audio file to 25 MB. OpenAI will use the `whisper-1` model by default.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **Audio**.
- **Operation:** Select **Transcribe a Recording**.
- **Input Data Field Name:** Defaults to data. Enter the name of the binary property that contains the audio file in one of these formats: `.flac`, `.mp3`, `.mp4`, `.mpeg`, `.mpga`, `.m4a`, `.ogg`, `.wav`, or `.webm`.

Options

- **Language of the Audio File:** Enter the language of the input audio in [ISO-639-1](#). Use this option to improve accuracy and latency.
- **Output Randomness (Temperature):** Defaults to 1.0. Adjust the randomness of the response. The range is between 0.0 (deterministic) and 1.0 (maximum randomness). We recommend altering this or **Output Randomness (Top P)** but not both. Start with a medium temperature (around 0.7) and adjust based on the outputs you observe. If the responses are too repetitive or rigid, increase the temperature. If they're too chaotic or off-track, decrease it.

Refer to [Create transcription | OpenAI](#) documentation for more information.

Translate a Recording

Use this operation to translate audio into English. OpenAI API limits the size of the audio file to 25 MB. OpenAI will use the `whisper-1` model by default.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **Audio**.
- **Operation:** Select **Translate a Recording**.
- **Input Data Field Name:** Defaults to data. Enter the name of the

binary property that contains the audio file in one of these formats: .flac, .mp3, .mp4, .mpeg, .mpga, .m4a, .ogg, .wav, or .webm.

Options

- **Output Randomness (Temperature):** Defaults to 1.0. Adjust the randomness of the response. The range is between 0.0 (deterministic) and 1.0 (maximum randomness). We recommend altering this or **Output Randomness (Top P)** but not both. Start with a medium temperature (around 0.7) and adjust based on the outputs you observe. If the responses are too repetitive or rigid, increase the temperature. If they're too chaotic or off-track, decrease it.

Refer to [Create transcription | OpenAI](#) documentation for more information.

Common issues

For common errors or issues and suggested resolution steps, refer to [Common Issues](#).

OpenAI Conversation operations

Use this operation to create, get, update, or remove a conversation in OpenAI. Refer to [OpenAI](#) for more information on the OpenAI node itself.

Create a Conversation

Use this operation to create a new conversation.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **Conversation**.
- **Operation:** Select **Create a Conversation**.
- **Messages:** A message input to the model. Messages with the system role take precedence over instructions given with the user role. Messages with the assistant role will be assumed to have been generated by the model in previous interactions.

Options

- **Metadata:** A set of key-value pairs for storing structured information. You can attach up to 16 pairs to an object, which is useful for adding custom data that can be used for searching via the API or in the dashboard.

Refer to [Conversations | OpenAI](#) documentation for more information.

Get a Conversation

Use this operation to retrieve an existing conversation.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **Conversation**.
- **Operation:** Select **Get Conversation**.
- **Conversation ID:** The ID of the conversation to retrieve.

Refer to [Conversations | OpenAI](#) documentation for more information.

Remove a Conversation

Use this operation to remove an existing conversation.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **Conversation**.
- **Operation:** Select **Remove Conversation**.
- **Conversation ID:** The ID of the conversation to remove.

Refer to [Conversations | OpenAI](#) documentation for more information.

Update a Conversation

Use this operation to update an existing conversation.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **Conversation**.
- **Operation:** Select **Update a Conversation**.
- **Conversation ID:** The ID of the conversation to update.

Options

- **Metadata:** A set of key-value pairs for storing structured information. You can attach up to 16 pairs to an object, which is useful for adding custom data that can be used for searching via the API or in the dashboard.

Refer to [Conversations | OpenAI](#) documentation for more information.

OpenAI File operations

Use this operation to create, delete, list, message, or update a file in OpenAI. Refer to [OpenAI](#) for more information on the OpenAI node itself.

Delete a File

Use this operation to delete a file from the server.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **File**.
- **Operation:** Select **Delete a File**.
- **File:** Enter the ID of the file to use for this operation or select the file name from the dropdown.

Refer to [Delete file | OpenAI](#) documentation for more information.

List Files

Use this operation to list files that belong to the user's organization.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **File**.
- **Operation:** Select **List Files**.

Options

- **Purpose:** Use this to only return files with the given purpose. Use **Assistants** to return only files related to Assistants and Message operations. Use **Fine-Tune** for files related to [Fine-tuning](#).

Refer to [List files | OpenAI](#) documentation for more information.

Upload a File

Use this operation to upload a file. This can be used across various operations.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **File**.
- **Operation:** Select **Upload a File**.
- **Input Data Field Name:** Defaults to `data`. Enter the name of the binary property which contains the file. The size of individual files can be a maximum of 512 MB or 2 million tokens for Assistants.

Options

- **Purpose:** Enter the intended purpose of the uploaded file. Use **Assistants** for files associated with Assistants and Message operations. Use **Fine-Tune** for [Fine-tuning](#).

Refer to [Upload file | OpenAI](#) documentation for more information.

Common issues

For common errors or issues and suggested resolution steps, refer to [Common Issues](#).

OpenAI Image operations

Use this operation to analyze or generate an image in OpenAI. Refer to [OpenAI](#) for more information on the OpenAI node itself.

Analyze Image

Use this operation to take in images and answer questions about them.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **Image**.
- **Operation:** Select **Analayze Image**.
- **Model:** Select the model you want to use to analyze an image.
- **Text Input:** Ask a question about the image.
- **Input Type:** Select how you'd like to input the image. Options include:
 - **Image URL(s):** Enter the **URL(s)** of the image(s) to analyze.
Add multiple URLs in a comma-separated list.
 - **Binary File(s):** Enter the name of the binary property which contains the image(s) in the **Input Data Field Name**.

Options

- **Detail:** Specify the balance between response time versus token usage.
- **Length of Description (Max Tokens):** Defaults to 300. Fewer tokens will result in shorter, less detailed image description.

Refer to [Images | OpenAI](#) documentation for more information.

Generate an Image

Use this operation to create an image from a text prompt.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **Image**.
- **Operation:** Select **Generate an Image**.
- **Model:** Select the model you want to use to generate an image.
- **Prompt:** Enter the text description of the desired image(s). The maximum length is 1000 characters for dall-e-2 and 4000 characters for dall-e-3.

Options

- **Quality:** The quality of the image you generate. **HD** creates images with finer details and greater consistency across the image. This option is only supported for dall-e-3. Otherwise, choose **Standard**.
- **Resolution:** Select the resolution of the generated images. Select **1024x1024** for dall-e-2. Select one of **1024x1024**, **1792x1024**, or **1024x1792** for dall-e-3 models.
- **Style:** Select the style of the generated images. This option is only supported for dall-e-3.
 - **Natural:** Use this to produce more natural looking images.
 - **Vivid:** Use this to produce hyper-real and dramatic images.
- **Respond with image URL(s):** Whether to return image URL(s) instead of binary file(s).
- **Put Output in Field:** Defaults to data. Enter the name of the output field to put the binary file data in. Only available if **Respond with image URL(s)** is turned off.

Refer to [Create image | OpenAI](#) documentation for more information.

Edit an Image

Use this operation to edit an image from a text prompt.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **Image**.
- **Operation:** Select **Edit Image**.
- **Model:** Select the model you want to use to generate an image. Supports dall-e-2 and gpt-image-1.
- **Prompt:** Enter the text description of the desired edits to the input image(s).
- **Image(s):** Add one or more binary fields to include images with your prompt. Each image should be a png, webp, or jpg file less than 50MB. You can provide up to 16 images.
- **Number of Images:** The number of images to generate. Must be between 1 and 10.
- **Size:** The size and dimensions of the generated images (in px).
- **Quality:** The quality of the image that will be generated (auto, low, medium, high, standard). Only supported for gpt-image-1.
- **Output Format:** The format in which the generated images are returned (png, webp, or jpg). Only supported for gpt-image-1.
- **Output Compression:** The compression level (0-100%) for the generated images. Only supported for gpt-image-1 with webp or jpeg output formats.

Options

- **Background:** Allows to set transparency for the background of the generated image(s). Only supported for gpt-image-1.
- **Input Fidelity:** Control how much effort the model will exert to match the style and features of input images. Only supported for gpt-image-1.
- **Image Mask:** Name of the binary property that contains the image. A second image whose fully transparent areas (for example, where alpha is zero) shows where the image should be edited. If there are multiple images provided, the mask will be applied on

the first image. Must be a valid PNG file, less than 4MB, and have the same dimensions as image.

- **User:** A unique identifier representing your end-user, which can help OpenAI to monitor and detect abuse.

Common issues

For common errors or issues and suggested resolution steps, refer to [Common Issues](#).

OpenAI Text operations

Use this operation to message a model or classify text for violations in OpenAI. Refer to [OpenAI](#) for more information on the OpenAI node itself.

Generate a Chat Completion

Use this operation to send a message or prompt to an OpenAI model - using the Chat Completions API - and receive a response.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **Text**.
- **Operation:** Select **Generate a Chat Completion**.
- **Model:** Select the model you want to use. If you're not sure which model to use, try gpt-4o if you need high intelligence or gpt-4o-mini if you need the fastest speed and lowest cost. Refer to [Models overview | OpenAI Platform](#) for more information.
- **Messages:** Enter a **Text** prompt and assign a **Role** that the model will use to generate responses. Refer to [Prompt engineering | OpenAI](#) for more information on how to write a better prompt by using these roles. Choose from one of these roles:
 - **User:** Sends a message as a user and gets a response from the model.
 - **Assistant:** Tells the model to adopt a specific tone or personality.
 - **System:** By default, there is no system message. You can define instructions in the user message, but the instructions set in the system message are more effective. You can set more than one system message per conversation. Use this to set the model's behavior or context for the next user message.
- **Simplify Output:** Turn on to return a simplified version of the response instead of the raw data.
- **Output Content as JSON:** Turn on to attempt to return the response in JSON format. Compatible with GPT-4 Turbo and all GPT-3.5 Turbo models newer than gpt-3.5-turbo-1106.

Options

- **Frequency Penalty:** Apply a penalty to reduce the model's tendency to repeat similar lines. The range is between 0.0 and 2.0.

- **Maximum Number of Tokens:** Set the maximum number of tokens for the response. One token is roughly four characters for standard English text. Use this to limit the length of the output.
- **Number of Completions:** Defaults to 1. Set the number of completions you want to generate for each prompt. Use carefully since setting a high number will quickly consume your tokens.
- **Presence Penalty:** Apply a penalty to influence the model to discuss new topics. The range is between 0.0 and 2.0.
- **Output Randomness (Temperature):** Adjust the randomness of the response. The range is between 0.0 (deterministic) and 1.0 (maximum randomness). We recommend altering this or **Output Randomness (Top P)** but not both. Start with a medium temperature (around 0.7) and adjust based on the outputs you observe. If the responses are too repetitive or rigid, increase the temperature. If they're too chaotic or off-track, decrease it. Defaults to 1.0.
- **Output Randomness (Top P):** Adjust the Top P setting to control the diversity of the assistant's responses. For example, 0.5 means half of all likelihood-weighted options are considered. We recommend altering this or **Output Randomness (Temperature)** but not both. Defaults to 1.0.

Refer to [Chat Completions | OpenAI](#) documentation for more information.

Generate a Model Response

Use this operation to send a message or prompt to an OpenAI model - using the Responses API - and receive a response.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [OpenAI credential](#).
- **Resource:** Select **Text**.
- **Operation:** Select **Generate a Model Response**.
- **Model:** Select the model you want to use. Refer to [Models overview | OpenAI Platform](#) for an overview.
- **Messages:** Choose from one of these a **Message Types**:
 - **Text:** Enter a **Text** prompt and assign a **Role** that the model will use to generate responses. Refer to [Prompt engineering | OpenAI](#) for more information on how to write a better prompt by using these roles.
 - **Image:** Provide an **Image** either through an Image URL, a File ID (using the [OpenAI Files API](#)) or by passing binary data from an earlier node in your workflow.
 - **File:** Provide a **File** in a supported format (currently: PDF only), either through a File URL, a File ID (using the [OpenAI Files API](#)) or by passing binary data from an earlier node in your workflow.
 - For any message type, you can choose from one of these roles:
 - **User:** Sends a message as a user and gets a response from the model.
 - **Assistant:** Tells the model to adopt a specific tone or personality.
 - **System:** By default, the system message is "You are a helpful assistant". You can define instructions in the user message, but the instructions set in the system message are

more effective. You can only set one system message per conversation. Use this to set the model's behavior or context for the next user message.

- **Simplify Output:** Turn on to return a simplified version of the response instead of the raw data.

Built-in Tools

The OpenAI Responses API provides a range of built-in tools to enrich the model's response:

- **Web Search:** Allows models to search the web for the latest information before generating a response.
- **MCP Servers:** Allows models to connect to remote MCP servers. Find out more about using remote MCP servers as tools [here](#).
- **File Search:** Allow models to search your knowledgebase from previously uploaded files for relevant information before generating a response. Refer to the [OpenAI documentation](#) for more information.
- **Code Interpreter:** Allows models to write and run Python code in a sandboxed environment.

Options

- **Maximum Number of Tokens:** Set the maximum number of tokens for the response. One token is roughly four characters for standard English text. Use this to limit the length of the output.
- **Output Randomness (Temperature):** Adjust the randomness of the response. The range is between 0.0 (deterministic) and 1.0 (maximum randomness). We recommend altering this or **Output Randomness (Top P)** but not both. Start with a medium temperature (around 0.7) and adjust based on the outputs you observe. If the responses are too repetitive or rigid, increase the temperature. If they're too chaotic or off-track, decrease it. Defaults to 1.0.
- **Output Randomness (Top P):** Adjust the Top P setting to control the diversity of the assistant's responses. For example, 0.5 means half of all likelihood-weighted options are considered. We recommend altering this or **Output Randomness (Temperature)** but not both. Defaults to 1.0.
- **Conversation ID:** The conversation that this response belongs to. Input items and output items from this response are automatically added to this conversation after this response completes.
- **Previous Response ID:** The ID of the previous response to continue from. Can't be used in conjunction with Conversation ID.
- **Reasoning:** The level of reasoning effort the model should spend to generate the response. Includes the ability to return a **Summary** of the reasoning performed by the model (for example, for debugging purposes).
- **Store:** Whether to store the generated model response for later retrieval via API. Defaults to true.
- **Output Format:** Whether to return the response as **Text**, in a specified **JSON Schema** or as a **JSON Object**.
- **Background:** Whether to run the model in background mode. This allows executing long-running tasks more reliably.

Refer to [Responses | OpenAI](#) documentation for more information.

Classify Text for Violations

Use this operation to identify and flag content that might be harmful. OpenAI model will analyze the text and return a response containing:

- **flagged**: A boolean field indicating if the content is potentially harmful.
- **categories**: A list of category-specific violation flags.
- **category_scores**: Scores for each category.

Enter these parameters:

- **Credential to connect with**: Create or select an existing [OpenAI credential](#).
- **Resource**: Select **Text**.
- **Operation**: Select **Classify Text for Violations**.
- **Text Input**: Enter text to classify if it violates the moderation policy.
- **Simplify Output**: Turn on to return a simplified version of the response instead of the raw data.

Options

- **Use Stable Model**: Turn on to use the stable version of the model instead of the latest version, accuracy may be slightly lower.

Refer to [Moderations | OpenAI](#) documentation for more information.

Common issues

For common errors or issues and suggested resolution steps, refer to [Common Issues](#).

OpenAI Video operations

Use this operation to generate a video in OpenAI. Refer to [OpenAI](#) for more information on the OpenAI node itself.

Generate Video

Use this operation to generate a video from a text prompt.

Enter these parameters:

- **Credential to connect with**: Create or select an existing [OpenAI credential](#).
- **Resource**: Select **Video**.
- **Operation**: Select **Generate Video**.
- **Model**: Select the model you want to use to generate a video. Currently supports sora-2 and sora-2-pro.
- **Prompt**: The prompt to generate a video from.
- **Seconds**: Clip duration in seconds (up to 25).
- **Size**: Output resolution formatted as width x height. 1024x1792 and 1792x1024 are only supported by Sora 2 Pro.

Options

- **Reference:** Optional image reference that guides generation. Has to be passed in as a binary item.
- **Wait Timeout:** Time to wait for the video to be generated in seconds. Defaults to 300.
- **Output Field Name:** The name of the output field to put the binary file data in. Defaults to data.

Refer to [Video Generation | OpenAI](#) for more information.

OpenAI node common issues

Here are some common errors and issues with the [OpenAI node](#) and steps to resolve or troubleshoot them.

-8<- “_snippets/integrations/openai-api-issues.md” -8<-
“_snippets/integrations/referenced-node-unexecuted.md”

OpenThesaurus node

Use the OpenThesaurus node to automate work in OpenThesaurus, and integrate OpenThesaurus with other applications. n8n supports synonym look-up for German words.

On this page, you'll find a list of operations the OpenThesaurus node supports and links to more resources.

Operations

- Get synonyms for a German word in German

Templates and examples

OpenWeatherMap node

Use the OpenWeatherMap node to automate work in OpenWeatherMap, and integrate OpenWeatherMap with other applications. n8n supports retrieving current and upcoming weather data with OpenWeatherMap.

On this page, you'll find a list of operations the OpenWeatherMap node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Returns the current weather data
- Returns the weather data for the next 5 days

Templates and examples

Oracle Database node

Use the Oracle Database node to automate work in Oracle Database, and integrate Oracle Database with other applications. n8n has built-in support for a wide range of Oracle Database features which includes executing an SQL statement, fetching, inserting, updating or deleting data from Oracle Database. This node uses the [node-oracledb driver](#) internally.

On this page, you'll find a list of operations the Oracle Database node supports and links to more resources.

-8<- "/_snippets/integrations/builtin/app-nodes/ai-tools.md"

Operations

- **Delete**: Delete an entire table or rows in a table
- **Execute SQL**: Execute an SQL statement
- **Insert**: Insert rows in a table
- **Insert or Update**: Insert or update rows in a table
- **Select**: Select rows from a table
- **Update**: Update rows in a table

Delete

Use this operation to delete an entire table or rows in a table.

Enter these parameters:

- **Credential to connect with**: Create or select an existing [Oracle Database credential](#).
- **Operation**: Select **Delete**.
- **Schema**: Choose the schema that contains the table you want to work on. Select **From list** to choose the schema from the dropdown list or **By Name** to enter the schema name.
- **Table**: Choose the table that you want to work on. Select **From list** to choose the table from the dropdown list, or select **By Name** to enter the table name.
- **Command**: The deletion action to take:
 - **Truncate**: Removes the table's data but preserves the table's structure.
 - **Delete**: Delete the rows that match the "Select Rows" condition. If you don't select anything, Oracle Database deletes all rows.
 - **Select Rows**: Define a **Column**, **Operator**, and **Value** to match rows on. The value can be passed as JSON using expression or string.
 - **Combine Conditions**: How to combine the conditions in "Select Rows". The **AND** requires all conditions to be true,

- while **OR** requires at least one condition to be true.
- **Drop:** Deletes the table's data and structure permanently.

Delete options

- **Auto Commit:** When this property is set to true, the transaction in the current connection is automatically committed at the end of statement execution.
- **Statement Batching:** The way to send statements to the database:
 - **Single Statement:** A single statement for all incoming items.
 - **Independently:** Execute one statement per incoming item of the execution.
 - **Transaction:** Execute all statements in a transaction. If a failure occurs, Oracle Database rolls back all changes.

Execute SQL

Use this operation to execute an SQL statement.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Oracle Database credential](#).
- **Operation:** Execute SQL **Execute SQL**.
- **Statement:** The SQL statement to execute. You can use n8n [expressions](#) and positional parameters like :1, :2, or named parameters like :name, :id to use with [Use bind parameters](#). To run a PL/SQL procedure, for example demo, you can use:

```
BEGIN
    demo;
END;
```

Execute Statement options

- **Auto Commit:** When this property is set to true, the transaction in the current connection is automatically committed at the end of statement execution.
- **Bind Variable Placeholder Values:** Enter the values for the bind parameters used in the statement [Use bind parameters](#).
- **Output Numbers As String:** Indicates if the numbers should be retrieved as a String.
- **Fetch Array Size:** This property is a number that sets the size of an internal buffer used for fetching query rows from Oracle Database. Changing it may affect query performance but does not affect how many rows are returned to the application.
- **Number of Rows to Prefetch:** This property is a query tuning option to set the number of additional rows the underlying Oracle driver fetches during the internal initial statement execution phase of a query.

Insert

Use this operation to insert rows in a table.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Oracle Database credential](#).
- **Operation:** Select **Insert**.
- **Schema:** Choose the schema that contains the table you want to work on. Select **From list** to choose the schema from the dropdown list or **By Name** to enter the schema name.
- **Table:** Choose the table that you want to work on. Select **From list** to choose the table from the dropdown list, or select **By Name** to enter the table name.
- **Mapping Column Mode:** How to map column names to incoming data:
 - **Map Each Column Manually:** Select the values to use for each column [Use n8n expressions for bind values](#).
 - **Map Automatically:** Automatically map incoming data to matching column names in Oracle Database. The incoming data field names must match the column names in Oracle Database for this to work. If necessary, consider using the [edit fields \(set\) node](#) before this node to adjust the format as needed.

Insert options

- **Auto Commit:** When this property is set to true, the transaction in the current connection is automatically committed at the end of statement execution.
- **Output Columns:** Choose which columns to output. You can select from a list of available columns or specify IDs using [expressions](#).
- **Statement Batching:** The way to send statements to the database:
 - **Single Statement:** A single statement for all incoming items.
 - **Independently:** Execute one statement per incoming item of the execution.
 - **Transaction:** Execute all statements in a transaction. If a failure occurs, Oracle Database rolls back all changes.

Insert or Update

Use this operation to insert or update rows in a table.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Oracle Database credential](#).
- **Operation:** Select **Insert or Update**.
- **Schema:** Choose the schema that contains the table you want to work on. Select **From list** to choose the schema from the dropdown list or **By Name** to enter the schema name.
- **Table:** Choose the table that you want to work on. Select **From list** to choose the table from the dropdown list, or select **By Name** to enter the table name.
- **Mapping Column Mode:** How to map column names to incoming data:
 - **Map Each Column Manually:** Select the values to use for each column [Use n8n expressions for bind values](#).
 - **Map Automatically:** Automatically map incoming data to matching column names in Oracle Database. The incoming data field names must match the column names in Oracle Database for this to work. If necessary, consider using the [edit fields \(set\) node](#) before this node to adjust the format as needed.

Insert or Update options

- **Auto Commit:** When this property is set to true, the transaction in the current connection is automatically committed at the end of statement execution.
- **Output Columns:** Choose which columns to output. You can select from a list of available columns or specify IDs using [expressions](#).
- **Statement Batching:** The way to send statements to the database:
 - **Single Statement:** A single statement for all incoming items.
 - **Independently:** Execute one statement per incoming item of the execution.
 - **Transaction:** Execute all statements in a transaction. If a failure occurs, Oracle Database rolls back all changes.

Select

Use this operation to select rows in a table.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Oracle Database credential](#).
- **Operation:** Select **Select**.
- **Schema:** Choose the schema that contains the table you want to work on. Select **From list** to choose the schema from the dropdown list or **By Name** to enter the schema name.
- **Table:** Choose the table that you want to work on. Select **From list** to choose the table from the dropdown list, or select **By Name** to enter the table name.
- **Return All:** Whether to return all results or only up to a given limit.
- **Limit:** The maximum number of items to return when **Return All** is disabled.
- **Select Rows:** Set the conditions to select rows. Define a **Column**, **Operator**, and **Value**(as json) to match rows on. The **Value** can vary by type — for example with Fixed mode:
 - String: "hello", helloworld, "hello with space"
 - Number: 12
 - JSON: { "key": "val" }

If you don't select anything, Oracle Database selects all rows. -

Combine Conditions: How to combine the conditions in **Select Rows**. The **AND** requires all conditions to be true, while **OR** requires at least one condition to be true. - **Sort:** Choose how to sort the selected rows. Choose a **Column** from a list or by ID and a sort **Direction**.

Select options

- **Auto Commit:** When this property is set to true, the transaction in the current connection is automatically committed at the end of statement execution.
- **Output Numbers As String:** Indicates if the numbers should be retrieved as a String.
- **Fetch Array Size:** This property is a number that sets the size of an internal buffer used for fetching query rows from Oracle Database. Changing it may affect query performance but does not

affect how many rows are returned to the application.

- **Number of Rows to Prefetch:** This property is a query tuning option to set the number of additional rows the underlying Oracle driver fetches during the internal initial statement execution phase of a query.

Update

Use this operation to update rows in a table.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Oracle Database credential](#).
- **Operation:** Select **Update**.
- **Schema:** Choose the schema that contains the table you want to work on. Select **From list** to choose the schema from the dropdown list or **By Name** to enter the schema name.
- **Table:** Choose the table that you want to work on. Select **From list** to choose the table from the dropdown list, or select **By Name** to enter the table name.
- **Mapping Column Mode:** How to map column names to incoming data:
 - **Map Each Column Manually:** Select the values to use for each column [Use n8n expressions for bind values](#).
 - **Map Automatically:** Automatically map incoming data to matching column names in Oracle Database. The incoming data field names must match the column names in Oracle Database for this to work. If necessary, consider using the [edit fields \(set\) node](#) before this node to adjust the format as needed.

Update options

- **Auto Commit:** When this property is set to true, the transaction in the current connection is automatically committed at the end of statement execution.
- **Output Columns:** Choose which columns to output. You can select from a list of available columns or specify IDs using [expressions](#).
- **Statement Batching:** The way to send statements to the database:
 - **Single Statement:** A single statement for all incoming items.
 - **Independently:** Execute one statement per incoming item of the execution.
 - **Transaction:** Execute all statements in a transaction. If a failure occurs, Oracle Database rolls back all changes.

Related resources

Refer to [SQL Language Reference](#) for more information about the service.

Refer to [node-oracledb documentation](#) for more information about the node-oracledb driver.

Use bind parameters

When creating a statement to run on an Oracle database instance, you can use the **Bind Variable Placeholder Values** field in the **Options** section to load data into the statement. n8n sanitizes data in statement parameters, which prevents SQL injection.

For example, you would want to find specific fruits by their color. Given the following input data:

```
[  
  {  
    "FRUIT_ID": 1,  
    "FRUIT_NAME": "Apple",  
    "COLOR": "Red"  
  },  
  {  
    "FRUIT_ID": 2,  
    "FRUIT_NAME": "Banana",  
    "COLOR": "Yellow"  
  }  
]
```

You can write a statement like:

```
SELECT * FROM FRUITS WHERE COLOR = :col
```

Then in **Bind Variable Placeholder Values**, provide the field values to use. You can provide fixed values or expressions. For this example, use expressions so the node can pull the color from each input item in turn:

```
// fruits is an example table name  
fruits, {{ $json.color }}
```

Use n8n Expressions for bind values

For **Values to Send**, you can provide inputs using n8n Expressions. Below are examples for different data types — you can either enter constant values or reference fields from previous items (\$json):

JSON

- Constant: {{ { k1: "v1", k2: "v2" } }}}
- From a previous item: {{ \$json.COL_JSON }}

VECTOR

- Constant: {{ [1, 2, 3, 4.5] }}
- From a previous item: {{ \$json.COL_VECTOR }}

BLOB

- Constant: {{ [94, 87, 34] }} or {{ 'BLOB data string' }}
- From a previous item: {{ \$json.COL_BLOB }}

RAW

- Constant: {{ [94, 87, 34] }}

- From a previous item: {{ \$json.COL_RAW }}

BOOLEAN

- Constant: {{ true }}
- From a previous item: {{ \$json.COL_BOOLEAN }}

NUMBER

- Constant: 1234
- From a previous item: {{ \$json.COL_NUMBER }}

VARCHAR

- Constant: 'Hello World'
- From a previous item: {{ \$json.COL_CHAR }}

These examples assume JSON keys (e.g. COL_JSON, COL_VECTOR) map directly to the respective SQL column types.

Oura node

Use the Oura node to automate work in Oura, and integrate Oura with other applications. n8n has built-in support for a wide range of Oura features, including getting profiles, and summaries.

On this page, you'll find a list of operations the Oura node supports and links to more resources.

Operations

- Profile
 - Get the user's personal information.
- Summary
 - Get the user's activity summary.
 - Get the user's readiness summary.
 - Get the user's sleep summary

Templates and examples

Paddle node

Use the Paddle node to automate work in Paddle, and integrate Paddle with other applications. n8n has built-in support for a wide range of Paddle features, including creating, updating, and getting coupons, as well as getting plans, products, and users.

On this page, you'll find a list of operations the Paddle node supports and links to more resources.

Operations

- Coupon
 - Create a coupon.
 - Get all coupons.
 - Update a coupon.
- Payment
 - Get all payment.
 - Reschedule payment.
- Plan
 - Get a plan.
 - Get all plans.
- Product
 - Get all products.
- User
 - Get all users

Templates and examples

PagerDuty node

Use the PagerDuty node to automate work in PagerDuty, and integrate PagerDuty with other applications. n8n has built-in support for a wide range of PagerDuty features, including creating incident notes, as well as updating, and getting all log entries and users.

On this page, you'll find a list of operations the PagerDuty node supports and links to more resources.

Operations

- Incident
 - Create an incident
 - Get an incident
 - Get all incidents
 - Update an incident
- Incident Note
 - Create an incident note
 - Get all incident's notes
- Log Entry
 - Get a log entry
 - Get all log entries
- User
 - Get a user

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

PayPal node

Use the PayPal node to automate work in PayPal, and integrate PayPal with other applications. n8n has built-in support for a wide range of PayPal features, including creating a batch payout and canceling unclaimed payout items.

On this page, you'll find a list of operations the PayPal node supports and links to more resources.

Operations

- Payout
 - Create a batch payout
 - Show batch payout details
- Payout Item
 - Cancels an unclaimed payout item
 - Show payout item details

Templates and examples

Peekalink node

Use the Peekalink node to automate work in Peekalink, and integrate Peekalink with other applications. n8n supports checking, and reviewing links with Peekalink.

On this page, you'll find a list of operations the Peekalink node supports and links to more resources.

Operations

- Check whether preview for a given link is available
- Return the preview for a link

Templates and examples

Perplexity node

Use the Perplexity node to automate work in Perplexity and integrate Perplexity with other applications. n8n has built-in support for messaging a model.

On this page, you'll find a list of operations the Perplexity node supports, and links to more resources.

Operations

- **Message a Model:** Create one or more completions for a given text.

Templates and examples

Related resources

Refer to [Perplexity's documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

PhantomBuster node

Use the PhantomBuster node to automate work in PhantomBuster, and integrate PhantomBuster with other applications. n8n has built-in support for a wide range of PhantomBuster features, including adding, deleting, and getting agents.

On this page, you'll find a list of operations the PhantomBuster node supports and links to more resources.

Operations

- Agent
 - Delete an agent by ID.
 - Get an agent by ID.
 - Get all agents of the current user's organization.
 - Get the output of the most recent container of an agent.
 - Add an agent to the launch queue.

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Philips Hue node

Use the Philips Hue node to automate work in Philips Hue, and integrate Philips Hue with other applications. n8n has built-in support for a wide range of Philips Hue features, including deleting, retrieving, and updating lights.

On this page, you'll find a list of operations the Philips Hue node supports and links to more resources.

Operations

- Light
 - Delete a light
 - Retrieve a light
 - Retrieve all lights
 - Update a light

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Pipedrive node

Use the Pipedrive node to automate work in Pipedrive, and integrate Pipedrive with other applications. n8n has built-in support for a wide range of Pipedrive features, including creating, updating, deleting, and getting activity, files, notes, organizations, and leads.

On this page, you'll find a list of operations the Pipedrive node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Activity
 - Create an activity
 - Delete an activity
 - Get data of an activity
 - Get data of all activities
 - Update an activity
- Deal
 - Create a deal
 - Delete a deal
 - Duplicate a deal
 - Get data of a deal
 - Get data of all deals
 - Search a deal
 - Update a deal
- Deal Activity
 - Get all activities of a deal
- Deal Product
 - Add a product to a deal
 - Get all products in a deal
 - Remove a product from a deal
 - Update a product in a deal
- File
 - Create a file
 - Delete a file
 - Download a file
 - Get data of a file
- Lead
 - Create a lead
 - Delete a lead
 - Get data of a lead

- Get data of all leads
- Update a lead
- Note
 - Create a note
 - Delete a note
 - Get data of a note
 - Get data of all notes
 - Update a note
- Organization
 - Create an organization
 - Delete an organization
 - Get data of an organization
 - Get data of all organizations
 - Update an organization
 - Search organizations
- Person
 - Create a person
 - Delete a person
 - Get data of a person
 - Get data of all persons
 - Search all persons
 - Update a person
- Product
 - Get data of all products

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Plivo node

Use the Plivo node to automate work in Plivo, and integrate Plivo with other applications. n8n has built-in support for a wide range of Plivo features, including making calls, and sending SMS/MMS.

On this page, you’ll find a list of operations the Plivo node supports and links to more resources.

Operations

- Call
 - Make a voice call
- MMS
 - Send an MMS message (US/Canada only)
- SMS
 - Send an SMS message.

Templates and examples

PostBin node

PostBin is a service that helps you test API clients and webhooks. Use the PostBin node to automate work in PostBin, and integrate PostBin with other applications. n8n has built-in support for a wide range of PostBin features, including creating and deleting bins, and getting and sending requests.

On this page, you'll find a list of operations the PostBin node supports, and links to more resources.

Operations

- Bin
 - Create
 - Get
 - Delete
- Request
 - Get
 - Remove First
 - Send

Templates and examples

Send requests

To send requests to a PostBin bin:

1. Go to [PostBin](#) and follow the steps to generate a new bin. PostBin gives you a unique URL, including a bin ID.
2. In the PostBin node, select the **Request** resource.
3. Choose the type of **Operation** you want to perform.
4. Enter your bin ID in **Bin ID**.

Create and manage bins

You can create and manage PostBin bins using the PostBin node.

1. In **Resource**, select **Bin**.
 2. Choose an **Operation**. You can create, delete, or get a bin.
-

Postgres node

Use the Postgres node to automate work in Postgres, and integrate Postgres with other applications. n8n has built-in support for a wide range of Postgres features, including executing queries, as well as inserting and updating rows in a database.

On this page, you'll find a list of operations the Postgres node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- **Delete**: Delete an entire table or rows in a table
- **Execute Query**: Execute an SQL query
- **Insert**: Insert rows in a table
- **Insert or Update**: Insert or update rows in a table
- **Select**: Select rows from a table
- **Update**: Update rows in a table

Delete

Use this operation to delete an entire table or rows in a table.

Enter these parameters:

- **Credential to connect with**: Create or select an existing [Postgres credential](#).
- **Operation**: Select **Delete**.
- **Schema**: Choose the schema that contains the table you want to work on. Select **From list** to choose the schema from the dropdown list or **By Name** to enter the schema name.
- **Table**: Choose the table that you want to work on. Select **From list** to choose the table from the dropdown list or **By Name** to enter the table name.
- **Command**: The deletion action to take:
 - **Truncate**: Removes the table's data but preserves the table's structure.
 - **Restart Sequences**: Whether to reset auto increment columns to their initial values as part of the Truncate process.
 - **Delete**: Delete the rows that match the "Select Rows" condition. If you don't select anything, Postgres deletes all rows.
 - **Select Rows**: Define a **Column**, **Operator**, and **Value** to match rows on.
 - **Combine Conditions**: How to combine the conditions in "Select Rows". **AND** requires all conditions to be true, while **OR** requires at least one condition to be true.
 - **Drop**: Deletes the table's data and structure permanently.

Delete options

- **Cascade**: Whether to also drop all objects that depend on the table, like views and sequences. Available if using **Truncate** or **Drop** commands.
- **Connection Timeout**: The number of seconds to try to connect to the database.
- **Delay Closing Idle Connection**: The number of seconds to wait before considering idle connections eligible for closing.
- **Query Batching**: The way to send queries to the database:
 - **Single Query**: A single query for all incoming items.
 - **Independently**: Execute one query per incoming item of the execution.
 - **Transaction**: Execute all queries in a transaction. If a failure occurs, Postgres rolls back all changes.
- **Output Large-Format Numbers As**: The format to output NUMERIC and BIGINT columns as:

- **Numbers:** Use this for standard numbers.
- **Text:** Use this if you expect numbers longer than 16 digits.
Without this, numbers may be incorrect.

Execute Query

Use this operation to execute an SQL query.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Postgres credential](#).
- **Operation:** Select **Execute Query**.
- **Query:** The SQL query to execute. You can use n8n [expressions](#) and tokens like \$1, \$2, and \$3 to build [prepared statements](#) to use with [query parameters](#).

Execute Query options

- **Connection Timeout:** The number of seconds to try to connect to the database.
- **Delay Closing Idle Connection:** The number of seconds to wait before considering idle connections eligible for closing.
- **Query Batching:** The way to send queries to the database:
 - **Single Query:** A single query for all incoming items.
 - **Independently:** Execute one query per incoming item of the execution.
 - **Transaction:** Execute all queries in a transaction. If a failure occurs, Postgres rolls back all changes.
- **Query Parameters:** A comma-separated list of values that you want to use as [query parameters](#).
- **Output Large-Format Numbers As:** The format to output NUMERIC and BIGINT columns as:
 - **Numbers:** Use this for standard numbers.
 - **Text:** Use this if you expect numbers longer than 16 digits.
Without this, numbers may be incorrect.
- **Replace Empty Strings with NULL:** Whether to replace empty strings with NULL in input. This may be useful when working with data exported from spreadsheet software.

Insert

Use this operation to insert rows in a table.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Postgres credential](#).
- **Operation:** Select **Insert**.
- **Schema:** Choose the schema that contains the table you want to work on. Select **From list** to choose the schema from the dropdown list or **By Name** to enter the schema name.
- **Table:** Choose the table that you want to work on. Select **From list** to choose the table from the dropdown list or **By Name** to enter the table name.
- **Mapping Column Mode:** How to map column names to incoming data:
 - **Map Each Column Manually:** Select the values to use for

each column.

- **Map Automatically:** Automatically map incoming data to matching column names in Postgres. The incoming data field names must match the column names in Postgres for this to work. If necessary, consider using the [edit fields \(set\) node](#) before this node to adjust the format as needed.

Insert options

- **Connection Timeout:** The number of seconds to try to connect to the database.
- **Delay Closing Idle Connection:** The number of seconds to wait before considering idle connections eligible for closing.
- **Query Batching:** The way to send queries to the database:
 - **Single Query:** A single query for all incoming items.
 - **Independently:** Execute one query per incoming item of the execution.
 - **Transaction:** Execute all queries in a transaction. If a failure occurs, Postgres rolls back all changes.
- **Output Columns:** Choose which columns to output. You can select from a list of available columns or specify IDs using [expressions](#).
- **Output Large-Format Numbers As:** The format to output NUMERIC and BIGINT columns as:
 - **Numbers:** Use this for standard numbers.
 - **Text:** Use this if you expect numbers longer than 16 digits. Without this, numbers may be incorrect.
- **Skip on Conflict:** Whether to skip the row if the insert violates a unique or exclusion constraint instead of throwing an error.
- **Replace Empty Strings with NULL:** Whether to replace empty strings with NULL in input. This may be useful when working with data exported from spreadsheet software.

Insert or Update

Use this operation to insert or update rows in a table.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Postgres credential](#).
- **Operation:** Select **Insert or Update**.
- **Schema:** Choose the schema that contains the table you want to work on. Select **From list** to choose the schema from the dropdown list or **By Name** to enter the schema name.
- **Table:** Choose the table that you want to work on. Select **From list** to choose the table from the dropdown list or **By Name** to enter the table name.
- **Mapping Column Mode:** How to map column names to incoming data:
 - **Map Each Column Manually:** Select the values to use for each column.
 - **Map Automatically:** Automatically map incoming data to matching column names in Postgres. The incoming data field names must match the column names in Postgres for this to work. If necessary, consider using the [edit fields \(set\) node](#) before this node to adjust the format as needed.

Insert or Update options

- **Connection Timeout:** The number of seconds to try to connect to the database.
- **Delay Closing Idle Connection:** The number of seconds to wait before considering idle connections eligible for closing.
- **Query Batching:** The way to send queries to the database:
 - **Single Query:** A single query for all incoming items.
 - **Independently:** Execute one query per incoming item of the execution.
 - **Transaction:** Execute all queries in a transaction. If a failure occurs, Postgres rolls back all changes.
- **Output Columns:** Choose which columns to output. You can select from a list of available columns or specify IDs using [expressions](#).
- **Output Large-Format Numbers As:** The format to output NUMERIC and BIGINT columns as:
 - **Numbers:** Use this for standard numbers.
 - **Text:** Use this if you expect numbers longer than 16 digits. Without this, numbers may be incorrect.
- **Replace Empty Strings with NULL:** Whether to replace empty strings with NULL in input. This may be useful when working with data exported from spreadsheet software.

Select

Use this operation to select rows in a table.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Postgres credential](#).
- **Operation:** Select **Select**.
- **Schema:** Choose the schema that contains the table you want to work on. Select **From list** to choose the schema from the dropdown list or **By Name** to enter the schema name.
- **Table:** Choose the table that you want to work on. Select **From list** to choose the table from the dropdown list or **By Name** to enter the table name.
- **Return All:** Whether to return all results or only up to a given limit.
- **Limit:** The maximum number of items to return when **Return All** is disabled.
- **Select Rows:** Set the conditions to select rows. Define a **Column**, **Operator**, and **Value** to match rows on. If you don't select anything, Postgres selects all rows.
- **Combine Conditions:** How to combine the conditions in **Select Rows**. **AND** requires all conditions to be true, while **OR** requires at least one condition to be true.
- **Sort:** Choose how to sort the selected rows. Choose a **Column** from a list or by ID and a sort **Direction**.

Select options

- **Connection Timeout:** The number of seconds to try to connect to the database.
- **Delay Closing Idle Connection:** The number of seconds to wait before considering idle connections eligible for closing.
- **Query Batching:** The way to send queries to the database:
 - **Single Query:** A single query for all incoming items.
 - **Independently:** Execute one query per incoming item of the

- execution.
- **Transaction:** Execute all queries in a transaction. If a failure occurs, Postgres rolls back all changes.
- **Output Columns:** Choose which columns to output. You can select from a list of available columns or specify IDs using [expressions](#).
- **Output Large-Format Numbers As:** The format to output NUMERIC and BIGINT columns as:
 - **Numbers:** Use this for standard numbers.
 - **Text:** Use this if you expect numbers longer than 16 digits. Without this, numbers may be incorrect.

Update

Use this operation to update rows in a table.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Postgres credential](#).
- **Operation:** Select **Update**.
- **Schema:** Choose the schema that contains the table you want to work on. Select **From list** to choose the schema from the dropdown list or **By Name** to enter the schema name.
- **Table:** Choose the table that you want to work on. Select **From list** to choose the table from the dropdown list or **By Name** to enter the table name.
- **Mapping Column Mode:** How to map column names to incoming data:
 - **Map Each Column Manually:** Select the values to use for each column.
 - **Map Automatically:** Automatically map incoming data to matching column names in Postgres. The incoming data field names must match the column names in Postgres for this to work. If necessary, consider using the [edit fields \(set\) node](#) before this node to adjust the format as needed.

Update options

- **Connection Timeout:** The number of seconds to try to connect to the database.
- **Delay Closing Idle Connection:** The number of seconds to wait before considering idle connections eligible for closing.
- **Query Batching:** The way to send queries to the database:
 - **Single Query:** A single query for all incoming items.
 - **Independently:** Execute one query per incoming item of the execution.
 - **Transaction:** Execute all queries in a transaction. If a failure occurs, Postgres rolls back all changes.
- **Output Columns:** Choose which columns to output. You can select from a list of available columns or specify IDs using [expressions](#).
- **Output Large-Format Numbers As:** The format to output NUMERIC and BIGINT columns as:
 - **Numbers:** Use this for standard numbers.
 - **Text:** Use this if you expect numbers longer than 16 digits. Without this, numbers may be incorrect.
- **Replace Empty Strings with NULL:** Whether to replace empty strings with NULL in input. This may be useful when working with data exported from spreadsheet software.

Templates and examples

Related resources

n8n provides a trigger node for Postgres. You can find the trigger node docs [here](#).

Use query parameters

When creating a query to run on a Postgres database, you can use the **Query Parameters** field in the **Options** section to load data into the query. n8n sanitizes data in query parameters, which prevents SQL injection.

For example, you want to find a person by their email address. Given the following input data:

```
[  
  {  
    "email": "alex@example.com",  
    "name": "Alex",  
    "age": 21  
  },  
  {  
    "email": "jamie@example.com",  
    "name": "Jamie",  
    "age": 33  
  }  
]
```

You can write a query like:

```
SELECT * FROM $1:name WHERE email = $2;
```

Then in **Query Parameters**, provide the field values to use. You can provide fixed values or expressions. For this example, use expressions so the node can pull the email address from each input item in turn:

```
// users is an example table name  
{{ [ 'users', $json.email ] }}
```

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

Postgres node common issues

Here are some common errors and issues with the [Postgres node](#) and steps to resolve or troubleshoot them.

Dynamically populate SQL IN groups with parameters

In Postgres, you can use the SQL IN comparison construct to make comparisons between groups of values:

```
SELECT color, shirt_size FROM shirts WHERE shirt_size IN ('small', 'medium', 'large');
```

While you can use n8n expressions in your query to dynamically populate the values in an IN group, combining this with query parameters provides extra protection by automatically sanitizing input.

To construct an IN group query with query parameters:

1. Set the **Operation** to **Execute Query**.
2. In **Options**, select **Query Parameters**.
3. Use an expression to select an array from the input data. For example, `{{ $json.input_shirt_sizes }}`.
4. In the **Query** parameter, write your query with the IN construct with an empty set of parentheses. For example: `sql SELECT color, shirt_size FROM shirts WHERE shirt_size IN ()`;
5. Inside of the IN parentheses, use an expression to dynamically create index-based placeholders (like `$1`, `$2`, and `$3`) for the number of items in your query parameter array. You can do this by increasing each array index by one since the placeholder variables are 1 indexed: `sql SELECT color, shirt_size FROM shirts WHERE shirt_size IN ({{ $json.input_shirt_sizes.map((i, pos) => "$" + (pos+1)).join(', ') }});`

With this technique, n8n automatically creates the correct number of prepared statement placeholders for the IN values according to the number of items in your array.

Working with timestamps and time zones

To avoid complications with how n8n and Postgres interpret timestamp and time zone data, follow these general tips:

- **Use UTC when storing and passing dates:** Using UTC helps avoid confusion over timezone conversions when converting dates between different representations and systems.
- **Set the execution timezone:** Set the global timezone in n8n using either environment variables (for self-hosted) or in the settings (for n8n Cloud). You can set a workflow-specific timezone in the workflow settings.
- **Use ISO 8601 format:** The ISO 8601 format encodes the day of the month, month, year, hour, minutes, and seconds in a standardized string. n8n passes dates between nodes as strings and uses Luxon to parse dates. If you need to cast to ISO 8601 explicitly, you can use the Date & Time node and a custom format set to the string `yyyy-MM-dd'T'HH:mm:ss`.

Outputting Date columns as date strings instead of ISO datetime strings

n8n's uses the [pg package](#) to integrate with Postgres, which affects how n8n processes date, timestamp, and related types from Postgres.

The pg package parses DATE values into new Date(row_value) by default, which produces a date that follows the [ISO 8601 datetime string](#) format. For example, a date of 2025-12-25 might produce a datetime string of 2025-12-25T23:00:00.000Z depending on the instance's timezone settings.

To work around this, use the [Postgres TO_CHAR function](#) to format the date into the expected format at query time:

```
SELECT TO_CHAR(date_col, 'YYYY-MM-DD') AS date_col_as_date FROM table_with_date_col
```

This will produce the date as a string without the time or timezone components. To continue the earlier example, with this casting, a date of 2025-12-25 would produce the string 2025-12-25. You can find out more in the [pg package documentation on dates](#).

PostHog node

Use the PostHog node to automate work in PostHog, and integrate PostHog with other applications. n8n has built-in support for a wide range of PostHog features, including creating aliases, events, and identity, as well as tracking pages.

On this page, you'll find a list of operations the PostHog node supports and links to more resources.

Operations

- Alias
 - Create an alias
- Event
 - Create an event
- Identity
 - Create
- Track
 - Track a page
 - Track a screen

Templates and examples

ProfitWell node

Use the ProfitWell node to automate work in ProfitWell, and integrate ProfitWell with other applications. n8n supports getting your company's account settings and retrieving financial metrics from ProfitWell.

On this page, you'll find a list of operations the ProfitWell node supports and links to more resources.

Operations

- Company
 - Get your company's ProfitWell account settings
- Metric
 - Retrieve financial metric broken down by day for either the current month or the last

Templates and examples

Pushbullet node

Use the Pushbullet node to automate work in Pushbullet, and integrate Pushbullet with other applications. n8n has built-in support for a wide range of Pushbullet features, including creating, updating, deleting, and getting a push.

On this page, you'll find a list of operations the Pushbullet node supports and links to more resources.

Operations

- Push
 - Create a push
 - Delete a push
 - Get all pushes
 - Update a push

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Pushcut node

Use the Pushcut node to automate work in Pushcut, and integrate Pushcut with other applications. n8n supports sending notifications with Pushcut.

On this page, you'll find a list of operations the Pushcut node supports and links to more resources.

Operations

- Notification
 - Send a notification

Templates and examples

Pushover node

Use the Pushover node to automate work in Pushover, and integrate Pushover with other applications. n8n supports sending push notifications with Pushover.

On this page, you'll find a list of operations the Pushover node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Message
 - Push

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

QuestDB node

Use the QuestDB node to automate work in QuestDB, and integrate QuestDB with other applications. n8n supports executing an SQL query and inserting rows in a database with QuestDB.

On this page, you'll find a list of operations the QuestDB node supports and links to more resources.

Operations

- Executes a SQL query.
- Insert rows in database.

Templates and examples

Node reference

Specify a column's data type

To specify a column's data type, append the column name with :type, where type is the data type you want for column. For example, if you want to specify the type int for the column **id** and type text for the column **name**, you can use the following snippet in the **Columns** field: id:int,name:text.

Quick Base node

Use the Quick Base node to automate work in Quick Base, and integrate Quick Base with other applications. n8n has built-in support for a wide range of Quick Base features, including creating, updating, deleting, and getting records, as well as getting fields, and downloading files.

On this page, you'll find a list of operations the Quick Base node supports and links to more resources.

Operations

- Field
 - Get all fields
- File
 - Delete a file
 - Download a file
- Record
 - Create a record
 - Delete a record
 - Get all records
 - Update a record
 - Upsert a record
- Report
 - Get a report
 - Run a report

Templates and examples

QuickBooks Online node

Use the QuickBooks node to automate work in QuickBooks, and integrate QuickBooks with other applications. n8n has built-in support for a wide range of QuickBooks features, including creating, updating, deleting, and getting bills, customers, employees, estimates, and invoices.

On this page, you'll find a list of operations the QuickBooks node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Bill
 - Create
 - Delete
 - Get
 - Get All
 - Update
- Customer
 - Create

- Get
- Get All
- Update
- Employee
 - Create
 - Get
 - Get All
 - Update
- Estimate
 - Create
 - Delete
 - Get
 - Get All
 - Send
 - Update
- Invoice
 - Create
 - Delete
 - Get
 - Get All
 - Send
 - Update
 - Void
- Item
 - Get
 - Get All
- Payment
 - Create
 - Delete
 - Get
 - Get All
 - Send
 - Update
 - Void
- Purchase
 - Get
 - Get All
- Transaction
 - Get Report
- Vendor
 - Create
 - Get
 - Get All
 - Update

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

QuickChart node

Use the QuickChart node to automate work in QuickChart, and integrate QuickChart with other applications. n8n has built-in support for a wide range of QuickChart chart types, including bar, doughnut, line, pie, and polar charts.

On this page, you'll find a list of operations the QuickChart node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

Create a chart by selecting the chart type:

- Chart Type
 - Bar Chart
 - Doughnut Chart
 - Line Chart
 - Pie Chart
 - Polar Chart

Templates and examples

Related resources

Refer to [QuickChart's API documentation](#) for more information about the service.

RabbitMQ node

Use the RabbitMQ node to automate work in RabbitMQ, and integrate RabbitMQ with other applications. n8n has built-in support for a wide range of RabbitMQ features, including accepting, and forwarding messages.

On this page, you'll find a list of operations the RabbitMQ node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Delete From Queue
- Send a Message to RabbitMQ

Templates and examples

Raindrop node

Use the Raindrop node to automate work in Raindrop, and integrate Raindrop with other applications. n8n has built-in support for a wide range of Raindrop features, including getting users, deleting tags, and creating, updating, deleting and getting collections and bookmarks.

On this page, you'll find a list of operations the Raindrop node supports and links to more resources.

Operations

- Bookmark
 - Create
 - Delete
 - Get
 - Get All
 - Update
- Collection
 - Create
 - Delete
 - Get
 - Get All
 - Update
- Tag
 - Delete
 - Get All
- User
 - Get

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Reddit node

Use the Reddit node to automate work in Reddit, and integrate Reddit with other applications. n8n has built-in support for a wide range of Reddit features, including getting profiles, and users, retrieving post comments and subreddit, as well as submitting, getting, and deleting posts.

On this page, you'll find a list of operations the Reddit node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Post
 - Submit a post to a subreddit
 - Delete a post from a subreddit
 - Get a post from a subreddit
 - Get all posts from a subreddit
 - Search posts in a subreddit or in all of Reddit.
- Post Comment
 - Create a top-level comment in a post
 - Retrieve all comments in a post
 - Remove a comment from a post
 - Write a reply to a comment in a post

- Profile
 - Get
- Subreddit
 - Retrieve background information about a subreddit.
 - Retrieve information about subreddits from all of Reddit.
- User
 - Get

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Redis node

Use the Redis node to automate work in Redis, and integrate Redis with other applications. n8n has built-in support for a wide range of Redis features, including deleting keys, getting key values, setting key value, and publishing messages to the Redis channel.

On this page, you'll find a list of operations the Redis node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Delete a key from Redis.
- Get the value of a key from Redis.
- Returns generic information about the Redis instance.
- Atomically increments a key by 1. Creates the key if it doesn't exist.
- Returns all the keys matching a pattern.
- Set the value of a key in Redis.
- Publish message to Redis channel.

Templates and examples

Rocket.Chat node

Use the Rocket.Chat node to automate work in Rocket.Chat, and integrate Rocket.Chat with other applications. n8n supports posting messages to channels, and sending direct messages, with Rocket.Chat.

On this page, you'll find a list of operations the Rocket.Chat node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Chat
 - Post a message to a channel or a direct message

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Rundeck node

Use the Rundeck node to automate work in Rundeck, and integrate Rundeck with other applications. n8n has built-in support for executing jobs and getting metadata.

On this page, you’ll find a list of operations the Rundeck node supports and links to more resources.

Operations

- **Job**
 - Execute a job
 - Get metadata of a job

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Find the job ID

1. Access your Rundeck dashboard.
 2. Open the project that contains the job you want to use with n8n.
 3. In the sidebar, select **JOBS**.
 4. Under **All Jobs**, select the name of the job you want to use with n8n.
 5. In the top left corner, under the name of the job, copy the string that’s displayed in smaller font below the job name. This is your job ID.
 6. Paste this job ID in the **Job Id** field in n8n.
-

S3 node

Use the S3 node to automate work in non-AWS S3 storage and integrate S3 with other applications. n8n has built-in support for a wide range of S3 features, including creating, deleting, and getting buckets, files, and folders. For AWS S3, use [AWS S3](#).

Use the S3 node for non-AWS S3 solutions like:

- [MinIO](#)
- [Wasabi](#)
- [Digital Ocean spaces](#)

On this page, you'll find a list of operations the S3 node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Bucket
 - Create a bucket
 - Delete a bucket
 - Get all buckets
 - Search within a bucket
- File
 - Copy a file
 - Delete a file
 - Download a file
 - Get all files
 - Upload a file
- Folder
 - Create a folder
 - Delete a folder
 - Get all folders

Templates and examples

Node reference

Setting file permissions in Wasabi

When uploading files to [Wasabi](#), you must set permissions for the files using the **ACL** dropdown and not the toggles.

[Image: File permissions when using the S3 node with Wasabi]

Salesforce node

Use the Salesforce node to automate work in Salesforce, and integrate Salesforce with other applications. n8n has built-in support for a wide range of Salesforce features, including creating, updating, deleting, and getting accounts, attachments, cases, and leads, as well as uploading documents.

On this page, you'll find a list of operations the Salesforce node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Account
 - Add note to an account
 - Create an account
 - Create a new account, or update the current one if it already exists (upsert)
 - Get an account
 - Get all accounts
 - Returns an overview of account's metadata.
 - Delete an account
 - Update an account
- Attachment
 - Create a attachment
 - Delete a attachment
 - Get a attachment
 - Get all attachments
 - Returns an overview of attachment's metadata.
 - Update a attachment
- Case
 - Add a comment to a case
 - Create a case
 - Get a case
 - Get all cases
 - Returns an overview of case's metadata
 - Delete a case
 - Update a case
- Contact
 - Add lead to a campaign
 - Add note to a contact
 - Create a contact
 - Create a new contact, or update the current one if it already exists (upsert)
 - Delete a contact
 - Get a contact
 - Returns an overview of contact's metadata
 - Get all contacts
 - Update a contact
- Custom Object
 - Create a custom object record
 - Create a new record, or update the current one if it already exists (upsert)
 - Get a custom object record
 - Get all custom object records
 - Delete a custom object record
 - Update a custom object record
- Document
 - Upload a document
- Flow
 - Get all flows
 - Invoke a flow
- Lead
 - Add lead to a campaign
 - Add note to a lead
 - Create a lead
 - Create a new lead, or update the current one if it already exists (upsert)
 - Delete a lead
 - Get a lead
 - Get all leads

- Returns an overview of Lead’s metadata
 - Update a lead
- Opportunity
 - Add note to an opportunity
 - Create an opportunity
 - Create a new opportunity, or update the current one if it already exists (upsert)
 - Delete an opportunity
 - Get an opportunity
 - Get all opportunities
 - Returns an overview of opportunity’s metadata
 - Update an opportunity
- Search
 - Execute a SOQL query that returns all the results in a single response
- Task
 - Create a task
 - Delete a task
 - Get a task
 - Get all tasks
 - Returns an overview of task’s metadata
 - Update a task
- User
 - Get a user
 - Get all users

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Working with Salesforce custom fields

To add custom fields to your request:

1. Select **Additional Fields > Add Field**.
2. In the dropdown, select **Custom Fields**.

You can then find and add your custom fields.

Salesmate node

Use the Salesmate node to automate work in Salesmate, and integrate Salesmate with other applications. n8n has built-in support for a wide range of Salesmate features, including creating, updating, deleting, and getting activities, companies, and deals.

On this page, you’ll find a list of operations the Salesmate node supports and links to more resources.

Operations

- Activity
 - Create an activity

- Delete an activity
- Get an activity
- Get all companies
- Update an activity
- Company
 - Create a company
 - Delete a company
 - Get a company
 - Get all companies
 - Update a company
- Deal
 - Create a deal
 - Delete a deal
 - Get a deal
 - Get all deals
 - Update a deal

Templates and examples

SeaTable node

Use the SeaTable node to automate work in SeaTable, and integrate SeaTable with other applications. n8n has built-in support for a wide range of SeaTable features, including creating, updating, deleting, updating, and getting rows.

On this page, you'll find a list of operations the SeaTable node supports and links to more resources.

Operations

- Row
 - Create
 - Delete
 - Get
 - Get All
 - Update

Templates and examples

SecurityScorecard node

Use the SecurityScorecard node to automate work in SecurityScorecard, and integrate SecurityScorecard with other applications. n8n has built-in support for a wide range of SecurityScorecard features, including creating, updating, deleting, and getting portfolio, as well as getting a company's data.

On this page, you'll find a list of operations the SecurityScorecard node supports and links to more resources.

Operations

- Company
 - Get company factor scores and issue counts
 - Get company's historical factor scores
 - Get company's historical scores
 - Get company information and summary of their scorecard
 - Get company's score improvement plan
- Industry
 - Get Factor Scores
 - Get Historical Factor Scores
 - Get Score
- Invite
 - Create an invite for a company/user
- Portfolio
 - Create a portfolio
 - Delete a portfolio
 - Get all portfolios
 - Update a portfolio
- Portfolio Company
 - Add a company to portfolio
 - Get all companies in a portfolio
 - Remove a company from portfolio
- Report
 - Download a generated report
 - Generate a report
 - Get list of recently generated report

Templates and examples

Segment node

Use the Segment node to automate work in Segment, and integrate Segment with other applications. n8n has built-in support for a wide range of Segment features, including adding users to groups, creating identities, and tracking activities.

On this page, you'll find a list of operations the Segment node supports and links to more resources.

Operations

- Group
 - Add a user to a group
- Identify
 - Create an identity
- Track
 - Record the actions your users perform. Every action triggers an event, which can also have associated properties.
 - Record page views on your website, along with optional extra information about the page being viewed.

Templates and examples

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

SendGrid node

Use the SendGrid node to automate work in SendGrid, and integrate SendGrid with other applications. n8n has built-in support for a wide range of SendGrid features, including creating, updating, deleting, and getting contacts, and lists, as well as sending emails.

On this page, you'll find a list of operations the SendGrid node supports and links to more resources.

```
-8<- "_snippets/integrations/builtin/app-nodes/ai-tools.md"
```

Operations

- Contact
 - Create/update a contact
 - Delete a contact
 - Get a contact by ID
 - Get all contacts
- List
 - Create a list
 - Delete a list
 - Get a list
 - Get all lists
 - Update a list
- Mail
 - Send an email.

Templates and examples

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

Sendy node

Use the Sendy node to automate work in Sendy, and integrate Sendy with other applications. n8n has built-in support for a wide range of Sendy features, including creating campaigns, and adding, counting, deleting, and getting subscribers.

On this page, you'll find a list of operations the Sendy node supports and links to more resources.

Operations

- Campaign
 - Create a campaign
- Subscriber

- Add a subscriber to a list
- Count subscribers
- Delete a subscriber from a list
- Unsubscribe user from a list
- Get the status of subscriber

Templates and examples

Sentry.io node

Use the Sentry.io node to automate work in Sentry.io, and integrate Sentry.io with other applications. n8n has built-in support for a wide range of Sentry.io features, including creating, updating, deleting, and getting, issues, projects, and releases, as well as getting all events.

On this page, you'll find a list of operations the Sentry.io node supports and links to more resources.

Operations

- Event
 - Get event by ID
 - Get all events
- Issue
 - Delete an issue
 - Get issue by ID
 - Get all issues
 - Update an issue
- Project
 - Create a new project
 - Delete a project
 - Get project by ID
 - Get all projects
 - Update a project
- Release
 - Create a release
 - Delete a release
 - Get release by version identifier
 - Get all releases
 - Update a release
- Organization
 - Create an organization
 - Get organization by slug
 - Get all organizations
 - Update an organization
- Team
 - Create a new team
 - Delete a team
 - Get team by slug
 - Get all teams
 - Update a team

Templates and examples

Related resources

Refer to [Sentry.io's documentation](#) for more information about the service.

-8<- “[snippets/integrations/builtin/app-nodes/operation-not-supported.md](#)”

ServiceNow node

Use the ServiceNow node to automate work in ServiceNow, and integrate ServiceNow with other applications. n8n has built-in support for a wide range of ServiceNow features, including getting business services, departments, configuration items, and dictionary as well as creating, updating, and deleting incidents, users, and table records.

On this page, you'll find a list of operations the ServiceNow node supports and links to more resources.

Operations

- Business Service
 - Get All
- Configuration Items
 - Get All
- Department
 - Get All
- Dictionary
 - Get All
- Incident
 - Create
 - Delete
 - Get
 - Get All
 - Update
- Table Record
 - Create
 - Delete
 - Get
 - Get All
 - Update
- User
 - Create
 - Delete
 - Get
 - Get All
 - Update
- User Group
 - Get All
- User Role
 - Get All

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

seven node

Use the seven node to automate work in seven, and integrate seven with other applications. n8n has built-in support for a wide range of seven features, including sending SMS, and converting text to voice.

On this page, you’ll find a list of operations the seven node supports and links to more resources.

Operations

- SMS
 - Send SMS
- Voice Call
 - Converts text to voice and calls a given number

Templates and examples

Shopify node

Use the Shopify node to automate work in Shopify, and integrate Shopify with other applications. n8n has built-in support for a wide range of Shopify features, including creating, updating, deleting, and getting orders and products.

On this page, you’ll find a list of operations the Shopify node supports and links to more resources.

Operations

- Order
 - Create an order
 - Delete an order
 - Get an order
 - Get all orders
 - Update an order
- Product
 - Create a product
 - Delete a product
 - Get a product
 - Get all products
 - Update a product

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

SIGNL4 node

Use the SIGNL4 node to automate work in SIGNL4, and integrate SIGNL4 with other applications. n8n supports sending and resolving alerts with SIGNL4.

On this page, you’ll find a list of operations the SIGNL4 node supports and links to more resources.

Operations

- Alert
 - Send an alert
 - Resolve an alert

Templates and examples

Slack node

Use the Slack node to automate work in Slack, and integrate Slack with other applications. n8n has built-in support for a wide range of Slack features, including creating, archiving, and closing channels, getting users and files, as well as deleting messages.

On this page, you’ll find a list of operations the Slack node supports and links to more resources.

Operations

- **Channel**
 - **Archive** a channel.
 - **Close** a direct message or multi-person direct message.
 - **Create** a public or private channel-based conversation.
 - **Get** information about a channel.
 - **Get Many**: Get a list of channels in Slack.
 - **History**: Get a channel’s history of messages and events.
 - **Invite** a user to a channel.
 - **Join** an existing channel.
 - **Kick**: Remove a user from a channel.
 - **Leave** a channel.
 - **Member**: List the members of a channel.
 - **Open** or resume a direct message or multi-person direct message.
 - **Rename** a channel.
 - **Replies**: Get a thread of messages posted to a channel.
 - **Sets purpose** of a channel.
 - **Sets topic** of a channel.

- **Unarchive** a channel.
- **File**
 - **Get** a file.
 - **Get Many**: Get and filter team files.
 - **Upload**: Create or upload an existing file.
- **Message**
 - **Delete** a message
 - **Get permalink**: Get a message's permalink.
 - **Search** for messages
 - **Send** a message
 - **Send and Wait for Approval**: Send a message and wait for approval from the recipient before continuing.
 - **Update** a message
- **Reaction**
 - **Add** a reaction to a message.
 - **Get** a message's reactions.
 - **Remove** a reaction from a message.
- **Star**
 - **Add** a star to an item.
 - **Delete** a star from an item.
 - **Get Many**: Get a list of an authenticated user's stars.
- **User**
 - **Get** information about a user.
 - **Get Many**: Get a list of users.
 - **Get User's Profile**.
 - **Get User's Status**.
 - **Update User's Profile**.
- **User Group**
 - **Create** a user group.
 - **Disable** a user group.
 - **Enable** a user group.
 - **Get Many**: Get a list of user groups.
 - **Update** a user group.

Templates and examples

Related resources

Refer to [Slack's documentation](#) for more information about the service.

Required scopes

Once you create a Slack app for your [Slack credentials](#), you must add the appropriate scopes to your Slack app for this node to work. Start with the scopes listed in the [Scopes | Slack credentials](#) page.

If those aren't enough, use the table below to look up the resource and operation you want to use, then follow the link to Slack's API documentation to find the correct scopes.

| Resource | Operation | Slack API method |
|----------|-----------|---------------------------------------|
| Channel | Archive | conversations.archive |
| Channel | Close | conversations.close |

| | | |
|------------|----------------------------|--|
| Channel | Create | conversations.create |
| Channel | Get | conversations.info |
| Channel | Get Many | conversations.list |
| Channel | History | conversations.history |
| Channel | Invite | conversations.invite |
| Channel | Join | conversations.join |
| Channel | Kick | conversations.kick |
| Channel | Leave | conversations.leave |
| Channel | Member | conversations.members |
| Channel | Open | conversations.open |
| Channel | Rename | conversations.rename |
| Channel | Replies | conversations.replies |
| Channel | Set Purpose | conversations.setPurpose |
| Channel | Set Topic | conversations.setTopic |
| Channel | Unarchive | conversations.unarchive |
| File | Get | files.info |
| File | Get Many | files.list |
| File | Upload | files.upload |
| Message | Delete | chat.delete |
| Message | Get Permalink | chat.getPermalink |
| Message | Search | search.messages |
| Message | Send | chat.postMessage |
| Message | Send and Wait for Approval | chat.postMessage |
| Message | Update | chat.update |
| Reaction | Add | reactions.add |
| Reaction | Get | reactions.get |
| Reaction | Remove | reactions.remove |
| Star | Add | stars.add |
| Star | Delete | stars.remove |
| Star | Get Many | stars.list |
| User | Get | users.info |
| User | Get Many | users.list |
| User | Get User's Profile | users.profile.get |
| User | Get User's Status | users.getPresence |
| User | Update User's Profile | users.profile.set |
| User Group | Create | usergroups.create |
| User Group | Disable | usergroups.disable |
| User Group | Enable | usergroups.enable |

| | | |
|------------|----------|--|
| User Group | Get Many | <u>usergroups.list</u> |
| User Group | Update | <u>usergroups.update</u> |

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Snowflake node

Use the Snowflake node to automate work in Snowflake, and integrate Snowflake with other applications. n8n has built-in support for a wide range of Snowflake features, including executing SQL queries, and inserting rows in a database.

On this page, you'll find a list of operations the Snowflake node supports and links to more resources.

Operations

- Execute an SQL query.
- Insert rows in database.
- Update rows in database.

Templates and examples

Splunk node

Use the Splunk node to automate work in Splunk, and integrate Splunk with other applications. n8n has built-in support for a wide range of Splunk features, including getting fired alerts reports, as well as deleting and getting search configuration.

On this page, you'll find a list of operations the Splunk node supports and links to more resources.

Operations

- Fired Alert
 - Get a fired alerts report
- Search Configuration
 - Delete a search configuration
 - Get a search configuration
 - Get many search configurations
- Search Job
 - Create a search job
 - Delete a search job
 - Get a search job
 - Get many search jobs
- Search Result

- Get many search results
- User
 - Create a user
 - Delete a user
 - Get a user
 - Get many users
 - Update a user

Templates and examples

Spotify node

Use the Spotify node to automate work in Spotify, and integrate Spotify with other applications. n8n has built-in support for a wide range of Spotify features, including getting album and artist information.

On this page, you'll find a list of operations the Spotify node supports and links to more resources.

Operations

- Album
 - Get an album by URI or ID.
 - Get a list of new album releases.
 - Get an album's tracks by URI or ID.
 - Search albums by keyword.
- Artist
 - Get an artist by URI or ID.
 - Get an artist's albums by URI or ID.
 - Get an artist's related artists by URI or ID.
 - Get an artist's top tracks by URI or ID.
 - Search artists by keyword.
- Library
 - Get the user's liked tracks.
- My Data
 - Get your followed artists.
- Player
 - Add a song to your queue.
 - Get your currently playing track.
 - Skip to your next track.
 - Pause your music.
 - Skip to your previous song.
 - Get your recently played tracks.
 - Resume playback on the current active device.
 - Set volume on the current active device.
 - Start playing a playlist, artist, or album.
- Playlist
 - Add tracks from a playlist by track and playlist URI or ID.
 - Create a new playlist.
 - Get a playlist by URI or ID.
 - Get a playlist's tracks by URI or ID.
 - Get a user's playlists.
 - Remove tracks from a playlist by track and playlist URI or ID.
 - Search playlists by keyword.

- Track
 - Get a track by its URI or ID.
 - Get audio features for a track by URI or ID.
 - Search tracks by keyword

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Stackby node

Use the Stackby node to automate work in Stackby, and integrate Stackby with other applications. n8n has built-in support for a wide range of Stackby features, including appending, deleting, listing and reading.

On this page, you'll find a list of operations the Stackby node supports and links to more resources.

Operations

- Append
- Delete
- List
- Read

Templates and examples

Storyblok node

Use the Storyblok node to automate work in Storyblok, and integrate Storyblok with other applications. n8n has built-in support for a wide range of Storyblok features, including getting, deleting, and publishing stories.

On this page, you'll find a list of operations the Storyblok node supports and links to more resources.

Operations

Content API

- Story
 - Get a story
 - Get all stories

Management API

- **Story**
 - Delete a story
 - Get a story
 - Get all stories
 - Publish a story
 - Unpublish a story

Templates and examples

Strapi node

Use the Strapi node to automate work in Strapi, and integrate Strapi with other applications. n8n has built-in support for a wide range of Strapi features, including creating and deleting entries.

On this page, you'll find a list of operations the Strapi node supports and links to more resources.

Operations

- Entry
 - Create
 - Delete
 - Get
 - Get Many
 - Update

Templates and examples

Related resources

Refer to [Strapi's documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Strava node

Use the Strava node to automate work in Strava, and integrate Strava with other applications. n8n has built-in support for a wide range of Strava features, including creating new activities, and getting activity information.

On this page, you'll find a list of operations the Strava node supports and links to more resources.

Operations

- Activity
 - Create a new activity
 - Get an activity
 - Get all activities
 - Get all activity comments
 - Get all activity kudos
 - Get all activity laps
 - Get all activity zones
 - Update an activity

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Stripe node

Use the Stripe node to automate work in Stripe, and integrate Stripe with other applications. n8n has built-in support for a wide range of Stripe features, including getting balance, creating charge and meter events, and deleting customers.

On this page, you'll find a list of operations the Stripe node supports and links to more resources.

Operations

- Balance
 - Get a balance
- Charge
 - Create a charge
 - Get a charge
 - Get all charges
 - Update a charge
- Coupon
 - Create a coupon
 - Get all coupons
- Customer
 - Create a customer
 - Delete a customer
 - Get a customer
 - Get all customers
 - Update a customer
- Customer Card
 - Add a customer card
 - Get a customer card
 - Remove a customer card
- Meter Event
 - Create a meter event
- Source
 - Create a source
 - Delete a source
 - Get a source
- Token
 - Create a token

Templates and examples

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

Supabase node

Use the Supabase node to automate work in Supabase, and integrate Supabase with other applications. n8n has built-in support for a wide range of Supabase features, including creating, deleting, and getting rows.

On this page, you'll find a list of operations the Supabase node supports and links to more resources.

```
-8<- "_snippets/integrations/builtin/app-nodes/ai-tools.md"
```

Operations

- Row
 - Create a new row
 - Delete a row
 - Get a row
 - Get all rows
 - Update a row

Using custom schemas

By default, the Supabase node only fetches the `public` schema. To fetch custom schemas, enable **Use Custom Schema**.

In the new **Schema** field, provide the custom schema the Supabase node should use.

Templates and examples

```
-8<- "_snippets/integrations/builtin/app-nodes/operation-not-supported.md"
```

Common issues

For common errors or issues and suggested resolution steps, refer to Common issues.

Supabase node common issues

Here are some common errors and issues with the Supabase node and steps to resolve or troubleshoot them.

Filtering rows by metadata

To filter rows by [Supabase metadata](#), set the **Select Type** to **String**.

From there, you can construct a query in the **Filters (String)** parameter to filter the metadata using the [Supabase metadata query language](#), inspired by the [MongoDB selectors](#) format. Access the metadata properties using the [Postgres -> arrow JSON operator](#) like this (curly brackets denote components to fill in):

```
metadata->>{your-property}={comparison-operator}.{comparison-value}
```

For example to access an age property in the metadata and return results greater than or equal to 21, you could enter the following in the **Filters (String)** field:

```
metadata->>age=gte.21
```

You can combine these operators to construct more complex queries.

Can't connect to a local Supabase database when using Docker

When you run Supabase in Docker, you need to configure the network so that n8n can connect to Supabase.

The solution depends on how you're hosting the two components.

If only Supabase is in Docker

If only Supabase is running in Docker, the Docker Compose file used by the [self-hosting guide](#) already runs Supabase bound to the correct interfaces.

When configuring [Supabase credentials](#), the localhost address should work without a problem (set the **Host** to `localhost`).

If Supabase and n8n are running in separate Docker containers

If both n8n and Supabase are running in Docker in separate containers, you can use Docker networking to connect them.

Configure Supabase to listen on all interfaces by binding to `0.0.0.0` inside of the container (the official [Docker compose configuration](#) already does this). Add both the Supabase and n8n components to the same [user-defined bridge network](#) if you aren't already managing them together in the same Docker Compose file.

When configuring [Supabase credentials](#), use the Supabase API gateway container's name (`supabase-kong` by default) as the host address instead of `localhost`. For example, if you use the default configuration, you would set the **Host** to `http://supabase-kong:8000`.

Records are accessible through Postgres but not Supabase

If queries for records return empty using the Supabase node, but are available through the [Postgres](#) node or with a Postgres client, there may be a conflict with Supabase's [Row Level Security \(RLS\)](#) policy.

Supabase always enables RLS when you create a table in a public schema with the Table Editor. When RLS is active, the API doesn't return any data with the public anon key until you create policies. This is a security measure to ensure that you only expose data you intend to.

To access data from a table with RLS enabled as the `anon` role, [create a policy](#) to enable the access patterns you intend to use.

SyncroMSP node

Use the SyncroMSP node to automate work in SyncroMSP, and integrate SyncroMSP with other applications. n8n has built-in support for a wide range of SyncroMSP features, including creating and deleting new customers, tickets, and contacts.

On this page, you'll find a list of operations the SyncroMSP node supports and links to more resources.

Operations

- Contact
 - Create new contact
 - Delete contact
 - Retrieve contact
 - Retrieve all contacts
 - Update contact
- Customer
 - Create new customer
 - Delete customer
 - Retrieve customer
 - Retrieve all customers
 - Update customer
- RMM
 - Create new RMM Alert
 - Delete RMM Alert
 - Retrieve RMM Alert
 - Retrieve all RMM Alerts
 - Mute RMM Alert
- Ticket
 - Create new ticket
 - Delete ticket
 - Retrieve ticket
 - Retrieve all tickets
 - Update ticket

Templates and examples

Taiga node

Use the Taiga node to automate work in Taiga, and integrate Taiga with other applications. n8n has built-in support for a wide range of Taiga features, including creating, updating, deleting, and getting issues.

On this page, you'll find a list of operations the Taiga node supports and links to more resources.

Operations

- Issue
 - Create an issue
 - Delete an issue
 - Get an issue
 - Get all issues
 - Update an issue

Templates and examples

Tapfiliate node

Use the Tapfiliate node to automate work in Tapfiliate, and integrate Tapfiliate with other applications. n8n has built-in support for a wide range of Tapfiliate features, including creating and deleting affiliates, and adding affiliate metadata.

On this page, you'll find a list of operations the Tapfiliate node supports and links to more resources.

Operations

- Affiliate
 - Create an affiliate
 - Delete an affiliate
 - Get an affiliate by ID
 - Get all affiliates
- Affiliate Metadata
 - Add metadata to affiliate
 - Remove metadata from affiliate
 - Update affiliate's metadata
- Program Affiliate
 - Add affiliate to program
 - Approve an affiliate for a program
 - Disapprove an affiliate
 - Get an affiliate in a program
 - Get all affiliates in program

Templates and examples

Telegram node

Use the Telegram node to automate work in [Telegram](#) and integrate Telegram with other applications. n8n has built-in support for a wide range of Telegram features, including getting files as well as deleting and editing messages.

On this page, you'll find a list of operations the Telegram node supports and links to more resources.

Operations

- [**Chat** operations](#)
 - [Get](#) up-to-date information about a chat.
 - [Get Administrators](#): Get a list of all administrators in a chat.
 - [Get Member](#): Get the details of a chat member.
 - [Leave](#) a chat.
 - [Set Description](#) of a chat.
 - [Set Title](#) of a chat.
 - [**Callback** operations](#)
 - [Answer Query](#): Send answers to callback queries sent from [inline keyboards](#).
 - [Answer Inline Query](#): Send answers to callback queries sent from [inline queries](#).
 - [**File** operations](#)
 - [Get File](#) from Telegram.
 - [**Message** operations](#)
 - [Delete Chat Message](#).
 - [Edit Message Text](#): Edit the text of an existing message.
 - [Pin Chat Message](#) for the chat.
 - [Send Animation](#) to the chat.
 - For use with GIFs or H.264/MPEG-4 AVC videos without sound up to 50 MB in size.
 - [Send Audio](#) file to the chat and display it in the music player.
 - [Send Chat Action](#): Tell the user that something is happening on the bot's side. The status is set for 5 seconds or less.
 - [Send Document](#) to the chat.
 - [Send Location](#): Send a geolocation to the chat.
 - [Send Media Group](#): Send a group of photos and/or videos.
 - [Send Message](#) to the chat.
 - [Send Photo](#) to the chat.
 - [Send Sticker](#) to the chat.
 - For use with static .WEBP, animated .TGS, or video .WEBM stickers.
 - [Send Video](#) to the chat.
 - [Unpin Chat Message](#) from the chat.
- ## Templates and examples

Related resources

Refer to [Telegram's API documentation](#) for more information about the service.

n8n provides a trigger node for Telegram. Refer to the trigger node docs [here](#) for more information.

Common issues

For common errors or issues and suggested resolution steps, refer to [Common Issues](#).

Telegram node Chat operations

Use these operations to get information about chats, members, administrators, leave chat, and set chat titles and descriptions. Refer to [Telegram](#) for more information on the Telegram node itself.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Get Chat

Use this operation to get up to date information about a chat using the Bot API [getChat](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Chat**.
- **Operation:** Select **Get**.
- **Chat ID:** Enter the Chat ID or username of the target channel in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.

Refer to the Telegram Bot API [getChat](#) documentation for more information.

Get Administrators

Use this operation to get a list of all administrators in a chat using the Bot API [getChatAdministrators](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Chat**.
- **Operation:** Select **Get Administrators**.
- **Chat ID:** Enter the Chat ID or username of the target channel in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.

Refer to the Telegram Bot API [getChatAdministrators](#) documentation for more information.

Get Chat Member

Use this operation to get the details of a chat member using the Bot API [getChatMember](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Chat**.
- **Operation:** Select **Get Member**.
- **Chat ID:** Enter the Chat ID or username of the target channel in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.
- **User ID:** Enter the unique identifier of the user whose information you want to get.

Refer to the Telegram Bot API [getChatMember](#) documentation for more information.

Leave Chat

Use this operation to leave a chat using the Bot API [leaveChat](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Chat**.
- **Operation:** Select **Leave**.
- **Chat ID:** Enter the Chat ID or username of the channel you wish to leave in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.

Refer to the Telegram Bot API [leaveChat](#) documentation for more information.

Set Description

Use this operation to set the description of a chat using the Bot API [setChatDescription](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Chat**.
- **Operation:** Select **Set Description**.
- **Chat ID:** Enter the Chat ID or username of the channel you wish to leave in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.
- **Description:** Enter the new description you'd like to set the chat to use, maximum of 255 characters.

Refer to the Telegram Bot API [setChatDescription](#) documentation for more information.

Set Title

Use this operation to set the title of a chat using the Bot API [setChatTitle](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Chat**.
- **Operation:** Select **Set Title**.
- **Chat ID:** Enter the Chat ID or username of the channel you wish to leave in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.
- **Title:** Enter the new title you'd like to set the chat to use, maximum of 255 characters.

Refer to the Telegram Bot API [setChatTitle](#) documentation for more information.

Telegram node Callback operations

Use these operations to respond to callback queries sent from the in-line keyboard or in-line queries. Refer to [Telegram](#) for more information on the Telegram node itself.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Answer Query

Use this operation to send answers to callback queries sent from [inline keyboards](#) using the Bot API [answerCallbackQuery](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Callback**.
- **Operation:** Select **Answer Query**.
- **Query ID:** Enter the unique identifier of the query you want to answer.
 - To feed a Query ID directly into this node, use the [Telegram Trigger](#) node triggered on the **Callback Query**.
- **Results:** Enter a JSON-serialized array of results you want to use as answers to the query. Refer to the Telegram [InlineQueryResults](#) documentation for more information on formatting your array.

Refer to the Telegram Bot API [answerCallbackQuery](#) documentation for more information.

Answer Query additional fields

Use the **Additional Fields** to further refine the behavior of the node. Select **Add Field** to add any of the following:

- **Cache Time:** Enter the maximum amount of time in seconds that the client may cache the result of the callback query. Telegram defaults to 0 seconds for this method.
- **Show Alert:** Telegram can display the answer as a notification at the top of the chat screen or as an alert. Choose whether you want to keep the default notification display (turned off) or display the answer as an alert (turned on).
- **Text:** If you want the answer to show text, enter up to 200 characters of text here.
- **URL:** Enter a URL that will be opened by the user's client. Refer to the **url** parameter instructions at the [Telegram Bot API answerCallbackQuery documentation](#) for more information.

Answer Inline Query

Use this operation to send answers to callback queries sent from inline queries using the Bot API [answerInlineQuery](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Callback**.
- **Operation:** Select **Answer Inline Query**.
- **Query ID:** Enter the unique identifier of the query you want to answer.
 - To feed a Query ID directly into this node, use the [Telegram Trigger](#) node triggered on the **Inline Query**.
- **Results:** Enter a JSON-serialized array of results you want to use as answers to the query. Refer to the [Telegram InlineQueryResults documentation](#) for more information on formatting your array.

Telegram allows a maximum of 50 results per query.

Refer to the [Telegram Bot API answerInlineQuery documentation](#) for more information.

Answer Inline Query additional fields

Use the **Additional Fields** to further refine the behavior of the node. Select **Add Field** to add any of the following:

- **Cache Time:** The maximum amount of time in seconds that the client may cache the result of the callback query. Telegram defaults to 300 seconds for this method.
- **Show Alert:** Telegram can display the answer as a notification at the top of the chat screen or as an alert. Choose whether you want to keep the default notification display (turned off) or display the answer as an alert (turned on).
- **Text:** If you want the answer to show text, enter up to 200 characters of text here.
- **URL:** Enter a URL that the user's client will open.

Telegram node File operations

Use this operation to get a file from Telegram. Refer to [Telegram](#) for more information on the Telegram node itself.

```
-8<- "_snippets/integrations/builtin/app-nodes/ai-tools.md"
```

Get File

Use this operation to get a file from Telegram using the Bot API [getFile](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **File**.
- **Operation:** Select **Get**.
- **File ID:** Enter the ID of the file you want to get.
- **Download:** Choose whether you want the node to download the file (turned on) or not (turned off).

Refer to the Telegram Bot API [getFile](#) documentation for more information.

Telegram node Message operations

Use these operations to send, edit, and delete messages in a chat; send files to a chat; and pin/unpin message from a chat. Refer to [Telegram](#) for more information on the Telegram node itself.

```
-8<- "_snippets/integrations/builtin/app-nodes/ai-tools.md" s ##  
Delete Chat Message
```

Use this operation to delete a message from chat using the Bot API [deleteMessage](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Message**.
- **Operation:** Select **Delete Chat Message**.
- **Chat ID:** Enter the Chat ID or username of the channel you wish to delete in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.
- **Message ID:** Enter the unique identifier of the message you want to delete.

Refer to the Telegram Bot API [deleteMessage](#) documentation for more information.

Edit Message Text

Use this operation to edit the text of an existing message using the Bot API [editMessageText](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Message**.
- **Operation:** Select **Edit Message Text**.
- **Chat ID:** Enter the Chat ID or username of the channel you wish to leave in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.
- **Message ID:** Enter the unique identifier of the message you want to edit.
- **Reply Markup:** Select whether to use the **Inline Keyboard** to display the [InlineKeyboardMarkup](#) **None** not to. This sets the `reply_markup` parameter. Refer to the [InlineKeyboardMarkup](#) documentation for more information.
- **Text:** Enter the text you want to edit the message to.

Refer to the Telegram Bot API [editMessageText](#) documentation for more information.

Edit Message Text additional fields

Use the **Additional Fields** to further refine the behavior of the node. Select **Add Field** to add any of the following:

- **Disable WebPage Preview:** Select whether you want to enable link previews for links in this message (turned off) or disable link previews for links in this message (turned on). This sets the `link_preview_options` parameter for `is_disabled`. Refer to the [LinkPreviewOptions](#) documentation for more information.
- **Parse Mode:** Choose whether the message should be parsed using **HTML** (default), **Markdown (Legacy)**, or **MarkdownV2**. This sets the `parse_mode` parameter.

Pin Chat Message

Use this operation to pin a message for the chat using the Bot API [pinChatMessage](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Message**.
- **Operation:** Select **Pin Chat Message**.
- **Chat ID:** Enter the Chat ID or username of the channel you wish to pin the message to in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.
- **Message ID:** Enter the unique identifier of the message you want to pin.

Refer to the Telegram Bot API [pinChatMessage](#) documentation for more information.

Pin Chat Message additional fields

Use the **Additional Fields** to further refine the behavior of the node. Select **Add Field** to add any of the following:

- **Disable Notifications:** By default, Telegram will notify all chat members that the message has been pinned. If you don't want these notifications to go out, turn this control on. Sets the `disable_notification` parameter to `true`.

Send Animation

Use this operation to send GIFs or H.264/MPEG-4 AVC videos without sound up to 50 MB in size to the chat using the Bot API [sendAnimation](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select [Message](#).
- **Operation:** Select [Send Animation](#).
- **Chat ID:** Enter the Chat ID or username of the channel you wish to send the animation to in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.
- **Binary File:** To send a binary file from the node itself, turn this option on. If you turn this parameter on, you must enter the **Input Binary Field** containing the file you want to send.
- **Animation:** If you aren't using the **Binary File**, enter the animation to send here. Pass a `file_id` to send a file that exists on the Telegram servers (recommended) or an HTTP URL for Telegram to get a file from the internet.
- **Reply Markup:** Use this parameter to set more interface options. Refer to [Reply Markup parameters](#) for more information on these options and how to use them.

Refer to the Telegram Bot API [sendAnimation](#) documentation for more information.

Send Animation additional fields

Use the **Additional Fields** to further refine the behavior of the node using optional fields in Telegram's `sendAnimation` method. Select **Add Field** to add any of the following:

- **Caption:** Enter a caption text for the animation, max of 1024 characters.
- **Disable Notification:** Choose whether to send the notification silently (turned on) or with a standard notification (turned off).
- **Duration:** Enter the animation's duration in seconds.
- **Height:** Enter the height of the animation.
- **Parse Mode:** Enter the parser to use for any related text. Options include **HTML** (default), **Markdown (Legacy)**, **MarkdownV2**.

Refer to Telegram's [Formatting options](#) for more information on these options.

- **Reply To Message ID:** If the message is a reply, enter the ID of the message it's replying to.
- **Message Thread ID:** Enter a unique identifier for the target message thread (topic) of the forum; for forum supergroups only.
- **Thumbnail:** Add the thumbnail of the file sent. Ignore this field if thumbnail generation for the file is supported server-side. The thumbnail should meet these specs:
 - JPEG format
 - Less than 200 KB in size
 - Width and height less than 320px.
- **Width:** Enter the width of the video clip.

Send Audio

Use this operation to send an audio file to the chat and display it in the music player using the Bot API [sendAudio](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Message**.
- **Operation:** Select **Send Audio**.
- **Chat ID:** Enter the Chat ID or username of the channel you wish to send the audio to in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.
- **Binary File:** To send a binary file from the node itself, turn this option on. If you turn this parameter on, you must enter the **Input Binary Field** containing the file you want to send.
- **Audio:** If you aren't using the **Binary File**, enter the audio to send here. Pass a `file_id` to send a file that exists on the Telegram servers (recommended) or an HTTP URL for Telegram to get a file from the internet.
- **Reply Markup:** Use this parameter to set more interface options. Refer to [Reply Markup parameters](#) for more information on these options and how to use them.

Refer to the Telegram Bot API [sendAudio](#) documentation for more information.

Send Audio additional fields

Use the **Additional Fields** to further refine the behavior of the node using optional fields in Telegram's `sendAudio` method. Select **Add Field** to add any of the following:

- **Caption:** Enter a caption text for the audio, max of 1024 characters.
- **Disable Notification:** Choose whether to send the notification silently (turned on) or with a standard notification (turned off).
- **Duration:** Enter the audio's duration in seconds.
- **Parse Mode:** Enter the parser to use for any related text. Options include **HTML** (default), **Markdown (Legacy)**, **MarkdownV2**. Refer to Telegram's [Formatting options](#) for more information on these options.

- **Performer:** Enter the name of the performer.
- **Reply To Message ID:** If the message is a reply, enter the ID of the message it's replying to.
- **Message Thread ID:** Enter a unique identifier for the target message thread (topic) of the forum; for forum supergroups only.
- **Title:** Enter the audio track's name.
- **Thumbnail:** Add the thumbnail of the file sent. Ignore this field if thumbnail generation for the file is supported server-side. The thumbnail should meet these specs:
 - JPEG format
 - Less than 200 KB in size
 - Width and height less than 320px.

Send Chat Action

Use this operation when you need to tell the user that something is happening on the bot's side. The status is set for 5 seconds or less using the Bot API [sendChatAction](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Message**.
- **Operation:** Select **Send Chat Action**.
- **Chat ID:** Enter the Chat ID or username of the channel you wish to send the chat action to in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.
- **Action:** Select the action you'd like to broadcast the bot as taking. The options here include: **Find Location**, **Typing**, **Recording** audio or video, and **Uploading** file types.

Refer to Telegram's Bot API [sendChatAction](#) documentation for more information.

Send Document

Use this operation to send a document to the chat using the Bot API [sendDocument](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Message**.
- **Operation:** Select **Send Document**.
- **Chat ID:** Enter the Chat ID or username of the channel you wish to send the document to in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.
- **Binary File:** To send a binary file from the node itself, turn this option on. If you turn this parameter on, you must enter the **Input Binary Field** containing the file you want to send.
- **Document:** If you aren't using the **Binary File**, enter the

document to send here. Pass a `file_id` to send a file that exists on the Telegram servers (recommended) or an HTTP URL for Telegram to get a file from the internet.

- **Reply Markup:** Use this parameter to set more interface options. Refer to [Reply Markup parameters](#) for more information on these options and how to use them.

Refer to Telegram's Bot API [sendDocument](#) documentation for more information.

Send Document additional fields

Use the **Additional Fields** to further refine the behavior of the node using optional fields in Telegram's `sendDocument` method. Select **Add Field** to add any of the following:

- **Caption:** Enter a caption text for the file, max of 1024 characters.
- **Disable Notification:** Choose whether to send the notification silently (turned on) or with a standard notification (turned off).
- **Parse Mode:** Enter the parser to use for any related text. Options include **HTML** (default), **Markdown (Legacy)**, **MarkdownV2**. Refer to [Formatting options](#) for more information on these options.
- **Reply To Message ID:** If the message is a reply, enter the ID of the message it's replying to.
- **Message Thread ID:** Enter a unique identifier for the target message thread (topic) of the forum; for forum supergroups only.
- **Thumbnail:** Add the thumbnail of the file sent. Ignore this field if thumbnail generation for the file is supported server-side. The thumbnail should meet these specs:
 - JPEG format
 - Less than 200 KB in size
 - Width and height less than 320px.

Send Location

Use this operation to send a geolocation to the chat using the Bot API [sendLocation](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Message**.
- **Operation:** Select **Send Location**.
- **Chat ID:** Enter the Chat ID or username of the channel you wish to send the location to in the format `@channelusername`.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.
- **Latitude:** Enter the latitude of the location.
- **Longitude:** Enter the longitude of the location.
- **Reply Markup:** Use this parameter to set more interface options. Refer to [Reply Markup parameters](#) for more information on these options and how to use them.

Refer to Telegram's Bot API [sendLocation](#) documentation for more information.

Send Location additional fields

Use the **Additional Fields** to further refine the behavior of the node using optional fields in Telegram's sendLocation method. Select **Add Field** to add any of the following:

- **Disable Notification:** Choose whether to send the notification silently (turned on) or with a standard notification (turned off).
- **Reply To Message ID:** If the message is a reply, enter the ID of the message it's replying to.
- **Message Thread ID:** Enter a unique identifier for the target message thread (topic) of the forum; for forum supergroups only.

Send Media Group

Use this operation to send a group of photos and/or videos using the Bot API [sendMediaGroup](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Message**.
- **Operation:** Select **Send Media Group**.
- **Chat ID:** Enter the Chat ID or username of the channel you wish to send the media group to in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.
- **Media:** Use **Add Media** to add different media types to your media group. For each medium, select:
 - **Type:** The type of media this is. Choose from **Photo** and **Video**.
 - **Media File:** Enter the media file to send. Pass a `file_id` to send a file that exists on the Telegram servers (recommended) or an HTTP URL for Telegram to get a file from the internet.
 - **Additional Fields:** For each media file, you can choose to add these fields:
 - **Caption:** Enter a caption text for the file, max of 1024 characters.
 - **Parse Mode:** Enter the parser to use for any related text. Options include **HTML** (default), **Markdown (Legacy)**, **MarkdownV2**. Refer to [Formatting options](#) for more information on these options.

Refer to Telegram's Bot API [sendMediaGroup](#) documentation for more information.

Send Media Group additional fields

Use the **Additional Fields** to further refine the behavior of the node using optional fields in Telegram's sendMediaGroup method. Select **Add Field** to add any of the following:

- **Disable Notification:** Choose whether to send the notification silently (turned on) or with a standard notification (turned off).
- **Reply To Message ID:** If the message is a reply, enter the ID of the message it's replying to.
- **Message Thread ID:** Enter a unique identifier for the target

message thread (topic) of the forum; for forum supergroups only.

Send Message

Use this operation to send a message to the chat using the Bot API [sendMessage](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Message**.
- **Operation:** Select **Send Message**.
- **Chat ID:** Enter the Chat ID or username of the channel you wish to send the message to in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.
- **Text:** Enter the text to send, max 4096 characters after entities parsing.

Refer to Telegram's Bot API [sendMessage](#) documentation for more information.

Send Message additional fields

Use the **Additional Fields** to further refine the behavior of the node using optional fields in Telegram's sendMessage method. Select **Add Field** to add any of the following:

- **Append n8n Attribution:** Choose whether to include the phrase This message was sent automatically with n8n to the end of the message (turned on, default) or not (turned off).
- **Disable Notification:** Choose whether to send the notification silently (turned on) or with a standard notification (turned off).
- **Disable WebPage Preview:** Select whether you want to enable link previews for links in this message (turned off) or disable link previews for links in this message (turned on). This sets the link_preview_options parameter for is_disabled. Refer to the [LinkPreviewOptions](#) documentation for more information.
- **Parse Mode:** Enter the parser to use for any related text. Options include **HTML** (default), **Markdown (Legacy)**, **MarkdownV2**. Refer to Telegram's [Formatting options](#) for more information on these options.
- **Reply To Message ID:** If the message is a reply, enter the ID of the message it's replying to.
- **Message Thread ID:** Enter a unique identifier for the target message thread (topic) of the forum; for forum supergroups only.

Send and Wait for Response

Use this operation to send a message to the chat using the Bot API [sendMessage](#) method and pause the workflow execution until the user confirms the operation.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Message**.
- **Operation:** Select **Send and Wait for Response**.
- **Chat ID:** Enter the Chat ID or username of the channel you wish to send the message to in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.
- **Message:** Enter the text to send.
- **Response Type:** The approval or response type to use:
 - **Approval:** Users can approve or disapprove from within the message.
 - **Free Text:** Users can submit a response with a form.
 - **Custom Form:** Users can submit a response with a custom form.

Refer to Telegram's Bot API [sendMessage](#) documentation for more information.

Send and Wait for Response additional fields

The additional fields depend on which **Response Type** you choose.

Approval

The **Approval** response type adds these options:

- **Type of Approval:** Whether to present only an approval button or both an approval and disapproval buttons.
- **Button Label:** The label for the approval or disapproval button. The default choice is Approve and Decline for approval and disapproval actions respectively.
- **Limit Wait Time:** Whether the workflow will automatically resume execution after a specified time limit. This can be an interval or a specific wall time.

Free Text

When using the Free Text response type, the following options are available:

- **Message Button Label:** The label to use for message button. The default choice is Respond.
- **Response Form Title:** The title of the form where users provide their response.
- **Response Form Description:** A description for the form where users provide their response.
- **Response Form Button Label:** The label for the button on the form to submit their response. The default choice is Submit.
- **Limit Wait Time:** Whether the workflow will automatically resume execution after a specified time limit. This can be an interval or a specific wall time.

Custom Form

When using the Custom Form response type, you build a form using the fields and options you want.

You can customize each form element with the settings outlined in the [n8n Form trigger's form elements](#). To add more fields, select the **Add Form Element** button.

The following options are also available:

- **Message Button Label:** The label to use for message button. The default choice is Respond.
- **Response Form Title:** The title of the form where users provide their response.
- **Response Form Description:** A description for the form where users provide their response.
- **Response Form Button Label:** The label for the button on the form to submit their response. The default choice is Submit.
- **Limit Wait Time:** Whether the workflow will automatically resume execution after a specified time limit. This can be an interval or a specific wall time.

Send Photo

Use this operation to send a photo to the chat using the Bot API [sendPhoto](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Message**.
- **Operation:** Select **Send Photo**.
- **Chat ID:** Enter the Chat ID or username of the channel you wish to send the photo to in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger node](#). Refer to [Common Issues | Get the Chat ID](#) for more information.
- **Binary File:** To send a binary file from the node itself, turn this option on. If you turn this parameter on, you must enter the **Input Binary Field** containing the file you want to send.
- **Photo:** If you aren't using the **Binary File**, enter the photo to send here. Pass a `file_id` to send a file that exists on the Telegram servers (recommended) or an HTTP URL for Telegram to get a file from the internet.
- **Reply Markup:** Use this parameter to set more interface options. Refer to [Reply Markup parameters](#) for more information on these options and how to use them.

Refer to Telegram's Bot API [sendPhoto](#) documentation for more information.

Send Photo additional fields

Use the **Additional Fields** to further refine the behavior of the node using optional fields in Telegram's `sendPhoto` method. Select **Add Field** to add any of the following:

- **Caption:** Enter a caption text for the file, max of 1024 characters.
- **Disable Notification:** Choose whether to send the notification silently (turned on) or with a standard notification (turned off).
- **Parse Mode:** Enter the parser to use for any related text. Options

include **HTML** (default), **Markdown (Legacy)**, **MarkdownV2**. Refer to Telegram's [Formatting options](#) for more information on these options.

- **Reply To Message ID:** If the message is a reply, enter the ID of the message it's replying to.
- **Message Thread ID:** Enter a unique identifier for the target message thread (topic) of the forum; for forum supergroups only.

Send Sticker

Use this method to send static .WEBP, animated .TGS, or video .WEBM stickers using the Bot API [sendSticker](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Message**.
- **Operation:** Select **Send Sticker**.
- **Chat ID:** Enter the Chat ID or username of the channel you wish to send the sticker to in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.
- **Binary File:** To send a binary file from the node itself, turn this option on. If you turn this parameter on, you must enter the **Input Binary Field** containing the file you want to send.
- **Sticker:** If you aren't using the **Binary File**, enter the photo to send here. Pass a `file_id` to send a file that exists on the Telegram servers (recommended) or an HTTP URL for Telegram to get a file from the internet.
- **Reply Markup:** Use this parameter to set more interface options. Refer to [Reply Markup parameters](#) for more information on these options and how to use them.

Refer to Telegram's Bot API [sendSticker](#) documentation for more information.

Send Sticker additional fields

Use the **Additional Fields** to further refine the behavior of the node using optional fields in Telegram's `sendSticker` method. Select **Add Field** to add any of the following:

- **Disable Notification:** Choose whether to send the notification silently (turned on) or with a standard notification (turned off).
- **Reply To Message ID:** If the message is a reply, enter the ID of the message it's replying to.
- **Message Thread ID:** Enter a unique identifier for the target message thread (topic) of the forum; for forum supergroups only.

Send Video

Use this operation to send a video to the chat using the Bot API [sendVideo](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Message**.
- **Operation:** Select **Send Video**.
- **Chat ID:** Enter the Chat ID or username of the channel you wish to send the video to in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.
- **Binary File:** To send a binary file from the node itself, turn this option on. If you turn this parameter on, you must enter the **Input Binary Field** containing the file you want to send.
- **Video:** If you aren't using the **Binary File**, enter the video to send here. Pass a `file_id` to send a file that exists on the Telegram servers (recommended) or an HTTP URL for Telegram to get a file from the internet.
- **Reply Markup:** Use this parameter to set more interface options. Refer to [Reply Markup parameters](#) for more information on these options and how to use them.

Refer to Telegram's Bot API [sendVideo](#) documentation for more information.

Send Video additional fields

Use the **Additional Fields** to further refine the behavior of the node using optional fields in Telegram's `sendVideo` method. Select **Add Field** to add any of the following:

- **Caption:** Enter a caption text for the video, max of 1024 characters.
- **Disable Notification:** Choose whether to send the notification silently (turned on) or with a standard notification (turned off).
- **Duration:** Enter the video's duration in seconds.
- **Height:** Enter the height of the video.
- **Parse Mode:** Enter the parser to use for any related text. Options include **HTML** (default), **Markdown (Legacy)**, **MarkdownV2**. Refer to Telegram's [Formatting options](#) for more information on these options.
- **Reply To Message ID:** If the message is a reply, enter the ID of the message it's replying to.
- **Message Thread ID:** Enter a unique identifier for the target message thread (topic) of the forum; for forum supergroups only.
- **Thumbnail:** Add the thumbnail of the file sent. Ignore this field if thumbnail generation for the file is supported server-side. The thumbnail should meet these specs:
 - JPEG format
 - Less than 200 KB in size
 - Width and height less than 320px.
- **Width:** Enter the width of the video.

Unpin Chat Message

Use this operation to unpin a message from the chat using the Bot API [unpinChatMessage](#) method.

Enter these parameters:

- **Credential to connect with:** Create or select an existing [Telegram credential](#).
- **Resource:** Select **Message**.
- **Operation:** Select **Pin Chat Message**.
- **Chat ID:** Enter the Chat ID or username of the channel you wish to unpin the message from in the format @channelusername.
 - To feed a Chat ID directly into this node, use the [Telegram Trigger](#) node. Refer to [Common Issues | Get the Chat ID](#) for more information.
- **Message ID:** Enter the unique identifier of the message you want to unpin.

Refer to the Telegram Bot API [unpinChatMessage](#) documentation for more information.

Reply Markup parameters

For most of the **Message Send** actions (such as Send Animation, Send Audio), use the **Reply Markup** parameter to set more interface options:

- **Force Reply:** The Telegram client will act as if the user has selected the bot's message and tapped **Reply**, automatically displaying a reply interface to the user. Refer to [Force Reply parameters](#) for further guidance on this option.
- **Inline Keyboard:** Display an inline keyboard right next to the message. Refer to [Inline Keyboard parameters](#) for further guidance on this option.
- **Reply Keyboard:** Display a custom keyboard with reply options. Refer to [Reply Keyboard parameters](#) for further guidance on this option.
- **Reply Keyboard Remove:** The Telegram client will remove the current custom keyboard and display the default letter-keyboard. Refer to [Reply Keyboard parameters](#) for further guidance on this option.

Force Reply parameters

Force Reply is useful if you want to create user-friendly step-by-step interfaces without having to sacrifice privacy mode.

If you select **Reply Markup > Force Reply**, choose from these **Force Reply** parameters:

- **Force Reply:** Turn on to show the reply interface to the user, as described above.
- **Selective:** Turn this on if you want to force reply from these users only:
 - Users that are @mentioned in the text of the message.
 - The sender of the original message, if this Send Animation message is a reply to a message.

Refer to [ForceReply](#) for more information.

Inline Keyboard parameters

If you select **Reply Markup > Inline Keyboard**, define the inline keyboard buttons you want to display using the **Add Button** option. To add more rows to your keyboard, use **Add Keyboard Row**.

Refer to [InlineKeyboardMarkup](#) and [InlineKeyboardButtons](#) for more information.

Reply Keyboard parameters

If you select **Reply Markup > Reply Keyboard**, use the **Reply Keyboard** section to define the buttons and rows in your Reply Keyboard.

Use the **Reply Keyboard Options** to further refine the keyboard's behavior:

- **Resize Keyboard:** Choose whether to request the Telegram client to resize the keyboard vertically for optimal fit (turned on) or whether to use the same height as the app's standard keyboard (turned off).
- **One Time Keyboard:** Choose whether the Telegram client should hide the keyboard as soon as a user uses it (turned on) or to keep displaying it (turned off).
- **Selective:** Turn this on if you want to show the keyboard to these users only:
 - Users that are @mentioned in the text of the message.
 - The sender of the original message, if this Send Animation message is a reply to a message.

Refer to [ReplyKeyboardMarkup](#) for more information.

Reply Keyboard Remove parameters

If you select **Reply Markup > Reply Keyboard Remove**, choose from these **Reply Keyboard Remove** parameters:

- **Remove Keyboard:** Choose whether to request the Telegram client to remove the custom keyboard (turned on) or to keep it (turned off).
- **Selective:** Turn this on if you want to remove the keyboard for these users only:
 - Users that are @mentioned in the text of the message.
 - The sender of the original message, if this Send Animation message is a reply to a message.

Refer to [ReplyKeyboardRemove](#) for more information.

Telegram node common issues

Here are some common errors and issues with the [Telegram node](#) and steps to resolve or troubleshoot them.

Add a bot to a Telegram channel

For a bot to send a message to a channel, you must add the bot to the channel. If you haven't added the bot to the channel, you'll see an error with a description like: Error: Forbidden: bot is not a participant of the channel.

To add a bot to a channel:

1. In the Telegram app, access the target channel and select the channel name.
2. Label the channel name as **public channel**.
3. Select **Administrators > Add Admin**.
4. Search for the bot's username and select it.
5. Select the checkmark on the top-right corner to add the bot to the channel.

Get the Chat ID

You can only use @channelusername on public channels. To interact with a Telegram group, you need that group's Chat ID.

There are three ways to get that ID:

1. From the Telegram Trigger: Use the [Telegram Trigger](#) node in your workflow to get a Chat ID. This node can trigger on different events and returns a Chat ID on successful execution.
2. From your web browser: Open Telegram in a web browser and open the group chat. The group's Chat ID is the series of digits behind the letter "g." Prefix your group Chat ID with a - when you enter it in n8n.
3. Invite Telegram's [@RawDataBot](#) to the group: Once you add it, the bot outputs a JSON file that includes a chat object. The id for that object is the group Chat ID. Then remove the RawDataBot from your group.

Send more than 30 messages per second

The Telegram API has a [limitation](#) of sending only 30 messages per second. Follow these steps to send more than 30 messages:

1. **Loop Over Items node:** Use the [Loop Over Items](#) node to get at most 30 chat IDs from your database.
2. **Telegram node:** Connect the Telegram node with the Loop Over Items node. Use the [Expression Editor](#) to select the Chat IDs from the Loop Over Items node.
3. **Code node:** Connect the [Code](#) node with the Telegram node. Use the Code node to wait for a few seconds before fetching the next batch of chat IDs. Connect this node with the Loop Over Items node.

You can also use this [workflow](#).

Remove the n8n attribution from sent messages

If you're using the node to [send Telegram messages](#), the message automatically gets an n8n attribution appended to the end:

| This message was sent automatically with n8n

To remove this attribution:

1. In the node's **Additional Fields** section, select **Add Field**.
2. Select **Append n8n attribution**.
3. Turn the toggle off.

Refer to [Send Message additional fields](#) for more information.

TheHive node

Use the TheHive node to automate work in TheHive, and integrate TheHive with other applications. n8n has built-in support for a wide range of TheHive features, including creating alerts, counting tasks logs, cases, and observables.

On this page, you'll find a list of operations the TheHive node supports and links to more resources.

Operations

The available operations depend on your API version. To see the operations list, create your credentials, including selecting your API version. Then return to the node, select the resource you want to use, and n8n displays the available operations for your API version.

- Alert
- Case
- Log
- Observable
- Task

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Related resources

n8n provides a trigger node for TheHive. You can find the trigger node docs [here](#).

Refer to TheHive's documentation for more information about the service:

- [Version 3](#)
 - [Version 4](#)
-

TheHive 5 node

Use the TheHive 5 node to automate work in TheHive, and integrate TheHive with other applications. n8n has built-in support for a wide range of TheHive features, including creating alerts, counting tasks logs, cases, and observables.

On this page, you'll find a list of operations the TheHive node supports and links to more resources.

Operations

- Alert
 - Create
 - Delete
 - Execute Responder
 - Get
 - Merge Into Case
 - Promote to Case
 - Search
 - Update
 - Update Status
- Case
 - Add Attachment
 - Create
 - Delete Attachment
 - Delete Case
 - Execute Responder
 - Get
 - Get Attachment
 - Get Timeline
 - Search
 - Update
- Comment
 - Create
 - Delete
 - Search
 - Update
- Observable
 - Create
 - Delete
 - Execute Analyzer
 - Execute Responder
 - Get
 - Search
 - Update
- Page
 - Create
 - Delete
 - Search
 - Update
- Query
 - Execute Query
- Task
 - Create
 - Delete
 - Execute Responder
 - Get
 - Search
 - Update
- Task Log

- Add Attachment
- Create
- Delete
- Delete Attachment
- Execute Responder
- Get
- Search

Templates and examples

Related resources

n8n provides a trigger node for TheHive. You can find the trigger node docs [here](#).

Refer to TheHive's [documentation](#) for more information about the service.

TimescaleDB node

Use the TimescaleDB node to automate work in TimescaleDB, and integrate TimescaleDB with other applications. n8n has built-in support for a wide range of TimescaleDB features, including executing an SQL query, as well as inserting and updating rows in a database.

On this page, you'll find a list of operations the TimescaleDB node supports and links to more resources.

Operations

- Execute an SQL query
- Insert rows in database
- Update rows in database

Templates and examples

Specify a column's data type

To specify a column's data type, append the column name with :type, where type is the data type you want for the column. For example, if you want to specify the type int for the column **id** and type text for the column **name**, you can use the following snippet in the **Columns** field: id:int,name:text.

Todoist node

Use the Todoist node to automate work in Todoist, and integrate Todoist with other applications. n8n has built-in support for a wide range of Todoist features, including creating, updating, deleting, and getting tasks.

On this page, you'll find a list of operations the Todoist node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Task
 - Create a new task
 - Close a task
 - Delete a task
 - Get a task
 - Get all tasks
 - Reopen a task
 - Update a task

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Travis CI node

Use the Travis CI node to automate work in Travis CI, and integrate Travis CI with other applications. n8n has built-in support for a wide range of Travis CI features, including cancelling and getting builds.

On this page, you'll find a list of operations the Travis CI node supports and links to more resources.

Operations

- Build
 - Cancel a build
 - Get a build
 - Get all builds
 - Restart a build
 - Trigger a build

Templates and examples

Trello node

Use the Trello node to automate work in Trello, and integrate Trello with other applications. n8n has built-in support for a wide range of Trello features, including creating and updating cards, and adding and removing members.

On this page, you'll find a list of operations the Trello node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Attachment
 - Create a new attachment for a card
 - Delete an attachment
 - Get the data of an attachment
 - Returns all attachments for the card
- Board
 - Create a new board
 - Delete a board
 - Get the data of a board
 - Update a board
- Board Member
 - Add
 - Get All
 - Invite
 - Remove
- Card
 - Create a new card
 - Delete a card
 - Get the data of a card
 - Update a card
- Card Comment
 - Create a comment on a card
 - Delete a comment from a card
 - Update a comment on a card
- Checklist
 - Create a checklist item
 - Create a new checklist
 - Delete a checklist
 - Delete a checklist item
 - Get the data of a checklist
 - Returns all checklists for the card
 - Get a specific checklist on a card
 - Get the completed checklist items on a card
 - Update an item in a checklist on a card
- Label
 - Add a label to a card.
 - Create a new label
 - Delete a label
 - Get the data of a label
 - Returns all labels for the board
 - Remove a label from a card.
 - Update a label.
- List
 - Archive/Unarchive a list
 - Create a new list
 - Get the data of a list
 - Get all the lists

- Get all the cards in a list
- Update a list

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Find the List ID

1. Open the Trello board that contains the list.
 2. If the list doesn't have any cards, add a card to the list.
 3. Open the card, add .json at the end of the URL, and press enter.
 4. In the JSON file, you will see a field called idList.
 5. Copy the contents of the idListfield and paste it in the ***List ID** field in n8n.
-

Twake node

Use the Twake node to automate work in Twake, and integrate Twake with other applications. n8n supports sending messages with Twake.

On this page, you'll find a list of operations the Twake node supports and links to more resources.

Operations

- Message
 - Send a message

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Twilio node

Use the Twilio node to automate work in Twilio, and integrate Twilio with other applications. n8n supports sending MMS/SMS and WhatsApp messages with Twilio.

On this page, you'll find a list of operations the Twilio node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- SMS
 - Send SMS/MMS/WhatsApp message
- Call
 - Make a phone call using text-to-speech to say a message

Templates and examples

Related resources

Refer to [Twilio's documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Twist node

Use the Twist node to automate work in Twist, and integrate Twist with other applications. n8n has built-in support for a wide range of Twist features, including creating conversations in a channel, as well as creating and deleting comments on a thread.

On this page, you'll find a list of operations the Twist node supports and links to more resources.

Operations

- Channel
 - Archive a channel
 - Initiates a public or private channel-based conversation
 - Delete a channel
 - Get information about a channel
 - Get all channels
 - Unarchive a channel
 - Update a channel
- Comment
 - Create a new comment to a thread
 - Delete a comment
 - Get information about a comment
 - Get all comments
 - Update a comment
- Message Conversation
 - Create a message in a conversation
 - Delete a message in a conversation
 - Get a message in a conversation
 - Get all messages in a conversation
 - Update a message in a conversation
- Thread
 - Create a new thread in a channel
 - Delete a thread
 - Get information about a thread
 - Get all threads
 - Update a thread

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Get the User ID

To get the User ID for a user:

1. Open the **Team** tab.
 2. Select a user's avatar.
 3. Copy the string of characters located after /u/ in your Twist URL.
This string is the User ID. For example, if the URL is
<https://twist.com/a/4qw45/people/u/475370> the User ID is 475370.
-

Unleashed Software node

Use the Unleashed Software node to automate work in Unleashed Software, and integrate Unleashed Software with other applications. n8n has built-in support for a wide range of Unleashed Software features, including getting sales orders and stock on hand.

On this page, you'll find a list of operations the Unleashed Software node supports and links to more resources.

Operations

- Sales Order
 - Get all sales orders
- Stock On Hand
 - Get a stock on hand
 - Get all stocks on hand

Templates and examples

UpLead node

Use the UpLead node to automate work in UpLead, and integrate UpLead with other applications. n8n supports several UpLead operations, including getting company information.

On this page, you'll find a list of operations the UpLead node supports and links to more resources.

Operations

- Company
 - Enrich
- Person

- Enrich

Templates and examples

uProc node

Use the uProc node to automate work in uProc, and integrate uProc with other applications. n8n has built-in support for a wide range of uProc features, including getting advanced human audio file, communication data, company, finance and product information.

On this page, you'll find a list of operations the uProc node supports and links to more resources.

Operations

Audio

- Get advanced human audio file by provided text and language
- Get an audio file by provided text and language

Communication

- Discover if a domain has a social network presence
- Discover if an email is valid, hard bounce, soft bounce, spam-trap, free, temporary, and recipient exists
- Discover if the email recipient exists, returning email status
- Check if an email domain has an SMTP server to receive emails
- Discover if the email has a social network presence
- Check if an email has a valid format
- Check if an email domain belongs to a disposable email service
- Check if email belongs to free service provider like Gmail
- Check if email is catchall
- Discover if an email exists in the Robinson list (only Spain)
- Check if email belongs to a system or role-based account
- Check if an email is a spam trap
- Discover if an IMEI number has a valid format
- Check if a LinkedIn profile is a first-degree contact
- Discover if mobile phone number exists in network operator, with worldwide coverage
- Discover if a mobile phone number has a valid format with worldwide coverage
- Discover if a mobile phone number has a valid format (only Spain)
- Discover if a mobile phone number has a valid prefix, with worldwide coverage
- Discover if a Spanish mobile phone number has a valid prefix
- Discover if a mobile number is switched on to call it later, with worldwide coverage
- Discover if a mobile number can receive SMS with worldwide coverage
- Discover if a phone (landline or mobile) exists in a Robinson list (only Spain)
- Discover if a landline or mobile number has a valid prefix
- Discover if a landline phone number is valid, with Spain coverage

- Allows discovering if landline number has a good international format, depending on the country
- Discover if a landline phone number prefix exists, with worldwide coverage
- Clean a phone removing non allowed characters
- Allows getting country code of a mobile phone number with international format
- Allows getting a domain from an email
- Discover an email by company website or domain and prospect's first-name and last-name
- Check if an email is personal or generic
- Get emails list found on the internet by domain or URI
- Get an emails list found on the internet by non-free email
- Get emails list found inside the website by domain or URI
- Get three first web references of an email published on the internet
- Allows you to fix the email domain of those misspelled emails
- Fix the international prefix of a phone based on the ISO code of a country
- Get GDPR compliant emails list by domain for your Email Marketing campaigns in Europe
- Discover if mobile exist using real-time HLR query
- Get personal email by social network profile
- Get portability data about a landline or mobile number, only for Spain
- Extract results from a LinkedIn search (employees in a company)
- Get members in a LinkedIn group
- Get 'Search LinkedIn Contacts' URL
- Extract the last 80 connections from your LinkedIn profile
- Extract the last 80 invitations sent from your LinkedIn
- Get users who comment on a post on LinkedIn
- Get users who like a post on LinkedIn
- Extract a LinkedIn profile
- Extract results from a LinkedIn search (profiles)
- Extract last profiles that have published content on LinkedIn by specific keywords
- Discover if mobile exist using real-time HLR query, as well as portability and roaming data
- Get existence, portability, and roaming of a mobile phone using MNP query
- Discover if mobile or landline prefix exists in Spain
- Allows normalizing email address, removing non allowed characters
- Allows normalizing a mobile phone, removing non-allowed characters
- Parse phone number in multiple fields and verify format and prefix validity
- Allows getting country prefix number by country code
- Discover an email by company website or domain and prospect's first-name and last-name
- This tool parses a social URI address and extracts any available indicators
- Search all social networks by domain, parses all found URLs, and returns social networks data
- Discover if a domain or a website has social activity and returns all social network profiles found
- Discover if an email has social activity, and get all social network profiles found
- Discover if a mobile phone has social activity, and get all social network profiles found

- Get web references for an email published on the internet
- Send a custom message invitation to a non connected LinkedIn profile
- Send a custom email to a recipient
- Send a custom SMS to a recipient with worldwide coverage
- Send a custom invitation message if a profile is connected or a custom message otherwise
- Visits a profile to show interest and get profile views in return from contact, increasing your LinkedIn network
- Send a custom private message to a connected LinkedIn profile
- Get an email by contact's LinkedIn profile URI
- Discover an email by company's name and prospect's full name
- Discover an email by company's website or domain and prospect's full name
- Get email by first name, last name, and company
- Get parsed and validated phone

Company

- Discover if a CIF card number is valid
- Check if a company is a debtor by TaxID
- Check if the ISIN number is valid
- Check if the SS number is valid, only for Spain
- Identify and classify a prospecting role in detecting the right area and seniority to filter later
- Get a company's contact, social, and technology data by domain
- Get a company's contact, social, and technology data by email
- Get a company's data by CIF
- Get a company's data by DUNS
- Get a company's data by domain
- Get a company's data by email
- Get a company's data by IP address
- Get a company's data by name
- Get a company's data by phone number
- Get a company's data by social networks URI (LinkedIn, Twitter)
- Get a company's name by company domain
- Get professional data of a decision-maker by company name/domain and area
- Discover more suitable decision-maker using search engines (Bing) by company name and area
- Get professional emails of decision-makers by company domain and area
- Discover up to ten decision-makers using search engines (Bing) by company name and area
- Get a company's domain by company name
- Get employees by company name or domain, area, seniority, and country
- Get a company's Facebook profile by name without manually searching on Google or Facebook
- Get geocoded company data by IP address
- Get a company's LinkedIn profile by name without manually searching on Google or LinkedIn
- Allows normalizing a CIF number, removing non-allowed characters
- Get a company's phone by company domain
- Get a company's sales data by a company's DUNS number
- Get a company's sales data by a company's domain name
- Get a company's sales data by a company's name
- Get a company's sales data by a company's tax ID (CIF)
- Get a company's Twitter profile by name without manually

- searching on Google or Twitter
- Get decision maker by search engine
- Get decision makers by search engine
- Get Facebook URI by company's domain
- Get GitHub URI by company's domain
- Get Instagram URI by company's domain
- Get LinkedIn URI by company's domain
- Get Pinterest URI by company's domain
- Get Twitter URI by company's domain
- Get YouTube URI by company's domain

Finance

- Check if crypto wallet is valid
- Discover if a BIC number has a valid format
- Discover if an account number has a valid format
- Check if credit card number checksum is valid
- Discover if an IBAN account number has a valid format
- Discover if an ISO currency code is valid
- Check if a TIN exists in Europe
- Convert amount between supported currencies and an exchange date
- Get credit card type
- Get multiple ISO currency codes by a country name
- Get all ISO currency by an IP address
- Get multiple ISO currency codes by a country ISO code
- Get ISO currency code by IP address
- Get ISO currency code by a currency ISO code
- Get ISO currency code by an ISO country code
- Get ISO currency code by a country name
- Get related European TIN in Europe
- Get IBAN by account number of the country
- Get to search data bank information by IBAN account number
- Get country VAT by address
- Get country VAT by coordinates
- Get Swift code lookup
- Get VAT by IP address
- Get VAT value by country ISO code
- Get VAT by phone number, with worldwide coverage
- Get VAT by zip code

Geographical

- Check if a country's ISO code exists
- Discover if the distance between two coordinates is equal to another
- Discover if the distance (kilometers) between two coordinates is greater than the given input
- Discover if the distance (kilometers) between two coordinates is greater or equal to the given input
- Discover if the distance(kilometers) between two coordinates is lower than the given input
- Check if an address exists by a partial address search
- Check if a house number exists by a partial address search
- Check if coordinates have a valid format
- Discover if a zip code number prefix exists (only for Spain)
- Discover if a zip code number has a valid format (only for Spain)
- Get cartesian coordinates(X, Y, Z/WGS84) by Latitude and Longitude

- Get location by parameters
- Get multiple cities by phone prefix (only for Spain)
- Get multiple cities by partial initial text (only for Spain)
- Get multiple cities by zip code prefix (only for Spain)
- Get a city from IP
- City search by partial name (only for Spain)
- Discover the city name by a local phone number (only for Spain)
- Discover the city name by the zip code (only for Spain)
- Discover the community name from a zip code (only for Spain)
- Discover latitude and longitude coordinates of an IP address
- Discover latitude and longitude coordinates of a postal address
- Get multiple country names by currency ISO code
- Get multiple countries by ISO code
- Get multiple country names by initial name
- Get country name by currency ISO code
- Get country name by IP address
- Get country name by its ISO code
- Get country by a prefix
- Get country name by phone number, with worldwide coverage
- Get Aplha2 code by a country prefix or a name
- Get decimal coordinates (degrees, minutes, and seconds) by latitude and longitude
- Returns straight-line distance (kilometers) between two addresses
- Returns straight-line distance (kilometers) between two GPS coordinates (latitude and longitude)
- Returns straight-line distance (kilometers) between two IP addresses
- Returns straight-line distance (kilometers) between two landline phones, using city and province of every phone
- Returns straight-line distance (kilometers) between two zip codes, using city and province of every zip code
- Get an exact address by a partial address search
- Discover geographical, company, timezone, and reputation data by IPv4 address
- Discover the city name, zip code, province, country, latitude, and longitude from an IPv4 or IPv6 address and geocodes it
- Parse postal address into separated fields, getting an improved resolution
- Discover locale data (currency, language) by IPv4 or IPv6 address
- Discover the city name, zip code, province, or country by latitude and longitude
- Discover the city name, zip code, province, country, latitude, and longitude from an IPv4 or IPv6 address
- Discover the city and the province from a landline phone number (only Spain)
- Discover location data by name
- Discover the city and the province from a zip code number (only Spain)
- Get the most relevant locations by name
- Get the most relevant locations by name, category, location, and radius
- Get multiple personal names by a prefix
- Discover network data by IPv4 or IPv6 address
- Allow normalizing an address by removing non allowed characters
- Allow normalizing a city by removing non allowed characters
- Allow normalizing a country by removing non allowed characters
- Allow normalizing a province by removing non allowed characters
- Allow normalizing a zip code by removing non allowed characters
- Get normalized country
- Parse postal address into separated fields, getting a basic

resolution

- Discover the province name from an IP address
- Get the first province by a name prefix (only for Spain)
- Discover the province name from a landline phone number (only for Spain)
- Discover the province name from a zip code number (only for Spain)
- Get a province list by a name prefix (only for Spain)
- Get a province list by a phone prefix (only for Spain)
- Get a province list by a zip code prefix (only for Spain)
- Discover reputation by IPv4 or IPv6 address
- Returns driving routing time, distance, fuel consumption, and cost between two addresses
- Returns driving routing time, distance, fuel consumption, and cost between two GPS coordinates
- Returns driving routing time, distance, fuel consumption, and cost between two IP addresses
- Returns driving routing time, distance, fuel consumption, and cost between two landline phones, using city and province of every phone (only for Spain)
- Returns driving routing time, distance, fuel consumption, and cost between two zip codes, using city and province of every zip code
- Discover date-time data by IPv4 or IPv6 address
- Get USNG coordinates by latitude and longitude
- Get UTM coordinates by latitude and longitude
- Discover the zip code if you have an IP address
- Get the first zip code by prefix, only for Spain
- Get multiple zip codes by prefix, with worldwide coverage
- Get time data by coordinates
- Get time data by postal address

Image

- Get QR code decoded content by an image URL
- It allows discovering all geographical and technical EXIF metadata present in a photographic JPEG image
- Get an encoded barcode by number and a required standard
- Get QR code encoded by a text
- Generate a new image by URL and text
- Discover logo (favicon) used in a domain
- Generate a screenshot by URL provided using Chrome browser
- Get OCR text from image

Internet

- Check if a domain exists
- Check if a domain has a DNS record
- Check if a domain has the given IP address assigned
- Check if a domain has an MX record
- Check if a domain has a valid SSL certificate
- Check if a domain has a valid format
- Check if a domain accepts all emails, existing or not
- Check if a domain is a free service domain provider
- Check if a domain is temporary or not
- Discover if a computer is switched on
- Discover if service in a port is available
- Check if an URL contains a string or regular expression
- Check if an URL exists
- Check that an URL has a valid format

- Get full SSL certificate data by a domain (or website) and monitor your certificate status
- Get feed entries by domain
- Get last feed entry by domain
- Get text data from web, PDF or image allowing to filter some elements by regular expressions or field names
- Decode URL to recover original
- Get valid, existing, and default URL when accessing a domain using a web browser
- Get long version of shortened URL
- Discover device features by a user agent
- Get the network name of and IP address
- Get the domain record by its type
- Encode URL to avoid problems
- Copy file from one URL to another URL
- Fix an IP address to the right format
- Get the IPv4 address linked with a domain
- Convert a number to an IP address
- Get ISP known name of email domain name
- Convert an IP address to numeric notation
- Scan a host and returns the most commonly open ports
- Obtains a list with multiple results from a website
- Obtains the content of a website
- Decode URL into multiple fields
- Generate a PDF file by URL (provided using Chrome browser)
- Get the root domain of any web address, removing non needed characters
- Generates shareable URIs to use on social networks and email using a content URI and a text
- Get data from the existing table in an HTML page or a PDF file
- Discover client and server technologies used in a domain
- Discover client and server technologies used in web pages
- Analyze URL's health status about SSL, broken links, conflictive HTTP links with SSL, and more
- Get website visits and rank of any domain
- Get a domain's WHOIS data by fields
- Get WHOIS data fields by IP address provided

Personal

- Check if age is between two numbers
- Check if date returns an age between 20 and 29
- Check if date returns an age between 40 and 49
- Check if age is greater than another
- Check if birth date returns an age greater than 64
- Check if birth date belongs to an adult (18 years for Spain)
- Check if age is lower than another
- Check if age is lower or equal than another
- Check if ages are equal
- Discover if a date is between two dates
- Discover if a date is greater
- Discover if a date is greater or equal
- Discover if a date belongs to a leap year
- Discover if a date is lower
- Discover if a date is lower or equal
- Discover if a date has a valid format
- Discover if a gender value is valid
- Discover if an NIE card number is valid
- Discover if a NIF card number is valid
- Check if a personal name exists in the INE data source (only for

Spain)

- Check if a name contains accepted characters
- Discover if a NIF exists in the Robinson list (only for Spain)
- Check if surname contains accepted characters
- Check if a personal surname appears in INE data source (only for Spain)
- Discover if a DNI card number is valid
- Discover the age of a birth date
- Discover the age range of a person by birth date
- Get the difference between two dates
- Discover the gender of a person by the email
- Discover the gender of a person or company by the name
- Get LinkedIn employee profile URI by business email
- Get LinkedIn employee profile URI by first name, last name, and company
- Discover the letter of a DNI card number
- Get first personal name matching by prefix and gender from INE data source (only for Spain)
- Get LinkedIn URI by email
- Get LinkedIn URI by phone
- Allow normalizing a DNI number by removing non allowed characters
- Allow normalizing an NIE number by removing non allowed characters
- Normalize name by removing non allowed characters
- Normalize surname
- Get parsed date-time
- Normalize full name, fixing abbreviations, sorting if necessary, and returning first name, last name, and gender
- Get prospect's contact data and the company's location and social data by email
- Get contact, location, and social data by email and company name and location
- Get personal and social data by social profile
- Get personal data by email
- Get personal data by first name, last name, company, and location
- Get personal data by mobile
- Get personal data by social network profile
- Generate random fake data
- Get first personal surname matching by prefix from INE data source (only for Spain)
- Get personal surname matching by prefix from INE data source (only for Spain)
- Get Twitter profile by first name, last name, and company
- Get XING profile by first name, last name, and company
- Add a contact email to a person list

Product

- Check if an ASIN code exists on the Amazon Marketplace
- Check if an ASIN code has a valid format
- Check if an EAN code exists on Amazon Marketplace
- Check if an EAN barcode has a valid format
- Check if an EAN barcode of 13 digits has a valid format
- Check if an EAN barcode of 14 digits has a valid format
- Check if an EAN barcode of 18 digits has a valid format
- Check if an EAN barcode of 8 digits has a valid format
- Check if a GTIN barcode has a valid format
- Check if a GTIN barcode of 13 digits has a valid format
- Check if a GTIN barcode of 14 digits has a valid format

- Check if a GTIN barcode of 8 digits has a valid format
- Check if VIN Number is valid
- Allows checking if an ISBN book exists
- Allows checking if an ISBN10/13 code has a valid format
- Allows checking if an ISBN10 code has a valid format
- Allows checking if an ISBN13 code has a valid format
- Check if a UPC exists
- Check if a UPC has a valid format
- Get ASIN by EAN
- Get a book by author's surname
- Get all publications by category
- Get book data by an editor's name
- Get book or publication data by 10 or 13 digits ISBN code
- Get book data by title
- Get books by author's surname
- Get all books by category
- Get all books by editor
- Get all books by title
- Get EAN code by ASIN code
- Get product data on a UPC on Amazon Marketplace
- Get ISBN10 code by ISBN13 code
- Get ISBN13 code by ISBN10 code
- Get data By VIN number

Security

- Check if a Luhn number is valid
- Check if a password is strong
- Check if a UUID number is valid
- Get blacklists for a domain
- Get blacklists for an IP address

Text

- Check if a string only contains alphabets
- Check if a string is alphanumeric
- Check if a string is boolean
- Check if the largest item in a list matches the provided item
- Check if IPv4 or IPv6 address has a valid format
- Check if IPv4 address has a valid format
- Check if IPv6 address has a valid format
- Check if the length of a list is between two quantities
- Checks if the length of a list equals a specified quantity
- Checks if the length of a list is greater than or equal to a certain amount
- Check if the length of a list is lower than a certain amount
- Check if the list contains a specific item
- Check if the list ends with a specific element
- Check if a list is sorted in ascending order
- Check if the list starts with a specific element
- Checks if the smallest element in a list matches the provided element
- Check if a string contains only numbers
- Check if a string contains a character
- Check if a string ends with a character
- Check if a string has no content
- Check if a string contains random characters
- Check if a string contains a value that matches with a regular expression

- Check if the length of a string is between two numbers
- Check if the length of a string is equal to a number
- Check if the length of a string is greater than a number
- Check if the length of a string is greater or equal to a number
- Check if the length of a string is lower than a number
- Check if the length of a string is lower or equal to a number
- Check if a string starts with a character
- Check if a string contains only lowercase characters
- Check if a string contains only uppercase characters
- Check if a list consists of unique elements
- Check if the supplied values form a valid list of elements
- Check if the number of words in a sentence is between two determined quantities
- Check if the number of words in a sentence equals a certain amount
- Check if the number of words in a sentence is greater than a certain amount
- Check if the number of words in a sentence is greater than
- Check if the word count is lower
- Check if the number of words present in a sentence is less than or equal to a quantity
- Convert a string to Base64 encoded value
- Discover banned English words in an email body or subject
- Get field names by analyzing the field value provided
- Get HTML code from Markdown
- Get Markdown text from HTML
- Get text without HTML
- Get spin string
- Format a string using a format pattern
- Generate random string using a regular expression as a pattern
- Return the largest item in a list
- Return the smallest item in a list
- Convert to lowercase
- Convert a string to MD5 encoded value
- Merge two strings
- Normalize a string depending on the field name
- Analyze string and return all emails, phones, zip codes, and links
- Convert a string to an SHA encoded value
- Analyze an English text with emojis and detect sentiment
- Returns an ascending sorted list
- Split a value into two parts and join them using a separator from the original string
- Split a value into two parts using a separator from the original string
- Get the length of a string
- Lookup string between multiple values by fuzzy logic and regex patterns
- Clean abuse words from a string
- Replace the first value found in a string with another
- Replace all values found in a string with another
- Translate a text into any language
- Return a single list with no repeating elements
- Convert all letters to uppercase
- Count total words in a text

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

UptimeRobot node

Use the UptimeRobot node to automate work in UptimeRobot, and integrate UptimeRobot with other applications. n8n has built-in support for a wide range of UptimeRobot features, including creating and deleting alerts, as well as getting account details.

On this page, you'll find a list of operations the UptimeRobot node supports and links to more resources.

Operations

- Account
 - Get account details
- Alert Contact
 - Create an alert contact
 - Delete an alert contact
 - Get an alert contact
 - Get all alert contacts
 - Update an alert contact
- Maintenance Window
 - Create a maintenance window
 - Delete a maintenance window
 - Get a maintenance window
 - Get all a maintenance windows
 - Update a maintenance window
- Monitor
 - Create a monitor
 - Delete a monitor
 - Get a monitor
 - Get all monitors
 - Reset a monitor
 - Update a monitor
- Public Status Page
 - Create a public status page
 - Delete a public status page
 - Get a public status page
 - Get all a public status pages

Templates and examples

urlscan.io node

Use the urlscan.io node to automate work in urlscan.io, and integrate urlscan.io with other applications. n8n has built-in support for a wide range of urlscan.io features, including getting and performing scans.

On this page, you'll find a list of operations the urlscan.io node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Scan
 - Get
 - Get All
 - Perform

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Venafi TLS Protect Cloud node

Use the Venafi TLS Protect Cloud node to automate work in Venafi TLS Protect Cloud, and integrate Venafi TLS Protect Cloud with other applications. n8n has built-in support for a wide range of Venafi TLS Protect Cloud features, including deleting and downloading certificates, as well as creating certificates requests.

On this page, you'll find a list of operations the Venafi TLS Protect Cloud node supports and links to more resources.

Operations

- Certificate
 - Delete
 - Download
 - Get
 - Get Many
 - Renew
- Certificate Request
 - Create
 - Get
 - Get Many

Templates and examples

Related resources

Refer to [Venafi's REST API documentation](#) for more information on this service.

n8n also provides:

- A [trigger node](#) for Venafi TLS Protect Cloud.
- A [node](#) for Venafi TLS Protect Datacenter.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Venafi TLS Protect Datacenter node

Use the Venafi TLS Protect Datacenter node to automate work in Venafi TLS Protect Datacenter, and integrate Venafi TLS Protect Datacenter with other applications. n8n has built-in support for a wide range of Venafi TLS Protect Datacenter features, including creating, deleting, and getting certificates.

On this page, you'll find a list of operations the Venafi TLS Protect Datacenter node supports and links to more resources.

Operations

- Certificate
 - Create
 - Delete
 - Download
 - Get
 - Get Many
 - Renew
- Policy
 - Get

Templates and examples

Related resources

n8n also provides:

- A node and trigger node for Venafi TLS Protect Cloud.
-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”
-

Vero node

Use the Vero node to automate work in Vero and integrate Vero with other applications. n8n has built-in support for a wide range of Vero features, including creating and deleting users.

On this page, you'll find a list of operations the Vero node supports and links to more resources.

Operations

- User
 - Create or update a user profile
 - Change a users identifier
 - Unsubscribe a user.
 - Resubscribe a user.

- Delete a user.
- Adds a tag to a users profile.
- Removes a tag from a users profile.
- Event
 - Track an event for a specific customer

Templates and examples

Vonage node

Use the Vonage node to automate work in Vonage, and integrate Vonage with other applications. n8n supports sending SMS with Vonage.

On this page, you'll find a list of operations the Vonage node supports and links to more resources.

Operations

- SMS
 - Send

Templates and examples

Webflow node

Use the Webflow node to automate work in Webflow, and integrate Webflow with other applications. n8n has built-in support for a wide range of Webflow features, including creating, updating, deleting, and getting items.

On this page, you'll find a list of operations the Webflow node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Item
 - Create
 - Delete
 - Get
 - Get All
 - Update

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Wekan node

Use the Wekan node to automate work in Wekan, and integrate Wekan with other applications. n8n has built-in support for a wide range of Wekan features, including creating, updating, deleting, and getting boards and cards.

On this page, you'll find a list of operations the Wekan node supports and links to more resources.

Operations

- Board
 - Create a new board
 - Delete a board
 - Get the data of a board
 - Get all user boards
- Card
 - Create a new card
 - Delete a card
 - Get a card
 - Get all cards
 - Update a card
- Card Comment
 - Create a comment on a card
 - Delete a comment from a card
 - Get a card comment
 - Get all card comments
- Checklist
 - Create a new checklist
 - Delete a checklist
 - Get the data of a checklist
 - Returns all checklists for the card
- Checklist Item
 - Delete a checklist item
 - Get a checklist item
 - Update a checklist item
- List
 - Create a new list
 - Delete a list
 - Get the data of a list
 - Get all board lists

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Load all the parameters for the node

To load all the parameters, for example, Author ID, you need to give admin permissions to the user. Refer to the [Wekan documentation](#) to learn how to change permissions.

WhatsApp Business Cloud node

Use the WhatsApp Business Cloud node to automate work in WhatsApp Business, and integrate WhatsApp Business with other applications. n8n has built-in support for a wide range of WhatsApp Business features, including sending messages, and uploading, downloading, and deleting media.

On this page, you'll find a list of operations the WhatsApp Business Cloud node supports and links to more resources.

Operations

- Message
 - Send
 - Send and Wait for Response
 - Send Template
- Media
 - Upload
 - Download
 - Delete

-8<- “_snippets/integrations/builtin/send-and-wait-operation.md”

Templates and examples

Related resources

Refer to [WhatsApp Business Platform's Cloud API documentation](#) for details about the operations.

Common issues

For common errors or issues and suggested resolution steps, refer to [Common Issues](#).

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

WhatsApp Business Cloud node common issues

Here are some common errors and issues with the [WhatsApp Business Cloud node](#) and steps to resolve or troubleshoot them.

Bad request - please check your parameters

This error occurs when WhatsApp Business Cloud rejects your request because of a problem with its parameters. It's common to see this when using the **Send Template** operation if the data you send doesn't match the format of your template.

To resolve this issue, review the parameters in your [message template](#). Pay attention to each parameter's data type and the order they're defined in the template.

Check the data that n8n is mapping to the template parameters. If you're using expressions to set parameter values, check the input data to make sure each item resolves to a valid value. You may want to use the [Edit Fields \(Set\) node](#) or [set a fallback value](#) to ensure you send a value with the correct format.

Working with non-text media

The WhatsApp Business Cloud node can work with non-text messages and media like images, audio, documents, and more.

If your operation includes a **Input Data Field Name** or **Property Name** parameter, set this to the field name itself rather than referencing the data in an expression.

For example, if you are trying to send a message with an "Image" **MessageType** and **Take Image From** set to "n8n", set **Input Data Field Name** to a field name like `data` instead of an expression like `{{ $json.input.data }}`.

Wise node

Use the Wise node to automate work in Wise, and integrate Wise with other applications. n8n has built-in support for a wide range of Wise features, including getting profiles, exchange rates, and recipients.

On this page, you'll find a list of operations the Wise node supports and links to more resources.

Operations

- Account
 - Retrieve balances for all account currencies of this user.
 - Retrieve currencies in the borderless account of this user.
 - Retrieve the statement for the borderless account of this user.
- Exchange Rate
 - Get
- Profile
 - Get
 - Get All
- Recipient
 - Get All

- Quote
 - Create
 - Get
- Transfer
 - Create
 - Delete
 - Execute
 - Get
 - Get All

Templates and examples

WooCommerce node

Use the WooCommerce node to automate work in WooCommerce, and integrate WooCommerce with other applications. n8n has built-in support for a wide range of WooCommerce features, including creating and deleting customers, orders, and products.

On this page, you'll find a list of operations the WooCommerce node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Customer
 - Create a customer
 - Delete a customer
 - Retrieve a customer
 - Retrieve all customers
 - Update a customer
- Order
 - Create a order
 - Delete a order
 - Get a order
 - Get all orders
 - Update an order
- Product
 - Create a product
 - Delete a product
 - Get a product
 - Get all products
 - Update a product

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

WordPress node

Use the WordPress node to automate work in WordPress, and integrate WordPress with other applications. n8n has built-in support for a wide range of WordPress features, including creating, updating, and getting posts and users.

On this page, you'll find a list of operations the WordPress node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Post
 - Create a post
 - Get a post
 - Get all posts
 - Update a post
- Pages
 - Create a page
 - Get a page
 - Get all pages
 - Update a page
- User
 - Create a user
 - Get a user
 - Get all users
 - Update a user

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

X (Formerly Twitter) node

Use the X node to automate work in X and integrate X with other applications. n8n has built-in support for a wide range of X features, including creating direct messages and deleting, searching, liking, and retweeting a tweet.

On this page, you'll find a list of operations the X node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Direct Message
 - Create a direct message
- Tweet
 - Create or reply a tweet
 - Delete a tweet
 - Search tweets
 - Like a tweet

- Retweet a tweet
- User
 - Get a user
- List
 - Add a member to a list

Templates and examples

Xero node

Use the Xero node to automate work in Xero, and integrate Xero with other applications. n8n has built-in support for a wide range of Xero features, including creating, updating, and getting contacts and invoices.

On this page, you'll find a list of operations the Xero node supports and links to more resources.

Operations

- Contact
 - Create a contact
 - Get a contact
 - Get all contacts
 - Update a contact
- Invoice
 - Create a invoice
 - Get a invoice
 - Get all invoices
 - Update a invoice

Templates and examples

Related resources

Refer to [Xero's API documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Yourls node

Use the Yourls node to automate work in Yourls, and integrate Yourls with other applications. n8n has built-in support for a wide range of Yourls features, including expanding and shortening URLs.

On this page, you'll find a list of operations the Yourls node supports and links to more resources.

Operations

- URL
 - Expand a URL
 - Shorten a URL
 - Get stats about one short URL

Templates and examples

YouTube node

Use the YouTube node to automate work in YouTube, and integrate YouTube with other applications. n8n has built-in support for a wide range of YouTube features, including retrieving and updating channels, as well as creating and deleting playlists.

On this page, you'll find a list of operations the YouTube node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Channel
 - Retrieve a channel
 - Retrieve all channels
 - Update a channel
 - Upload a channel banner
- Playlist
 - Create a playlist
 - Delete a playlist
 - Get a playlist
 - Retrieve all playlists
 - Update a playlist
- Playlist Item
 - Add an item to a playlist
 - Delete a item from a playlist
 - Get a playlist's item
 - Retrieve all playlist items
- Video
 - Delete a video
 - Get a video
 - Retrieve all videos
 - Rate a video
 - Update a video
 - Upload a video
- Video Category
 - Retrieve all video categories

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Zammad node

Use the Zammad node to automate work in Zammad, and integrate Zammad with other applications. n8n has built-in support for a wide range of Zammad features, including creating, retrieving, and deleting groups and organizations.

On this page, you'll find a list of operations the Zammad node supports and links to more resources.

Operations

- Group
 - Create
 - Delete
 - Get
 - Get many
 - Update
- Organization
 - Create
 - Delete
 - Get
 - Get many
 - Update
- Ticket
 - Create
 - Delete
 - Get
 - Get many
- User
 - Create
 - Delete
 - Get
 - Get many
 - Get self
 - Update

Templates and examples

Zendesk node

Use the Zendesk node to automate work in Zendesk, and integrate Zendesk with other applications. n8n has built-in support for a wide range of Zendesk features, including creating, and deleting tickets, users, and organizations.

On this page, you'll find a list of operations the Zendesk node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Ticket
 - Create a ticket
 - Delete a ticket
 - Get a ticket
 - Get all tickets
 - Recover a suspended ticket
 - Update a ticket
- Ticket Field
 - Get a ticket field
 - Get all system and custom ticket fields
- User
 - Create a user
 - Delete a user
 - Get a user
 - Get all users
 - Get a user's organizations
 - Get data related to the user
 - Search users
 - Update a user
- Organization
 - Create an organization
 - Delete an organization
 - Count organizations
 - Get an organization
 - Get all organizations
 - Get data related to the organization
 - Update a organization

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Zoho CRM node

Use the Zoho CRM node to automate work in Zoho CRM, and integrate Zoho CRM with other applications. n8n has built-in support for a wide range of Zoho CRM features, including creating and deleting accounts, contacts, and deals.

On this page, you'll find a list of operations the Zoho CRM node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Account
 - Create an account
 - Create a new record, or update the current one if it already exists (upsert)
 - Delete an account
 - Get an account

- Get all accounts
 - Update an account
- Contact
 - Create a contact
 - Create a new record, or update the current one if it already exists (upsert)
 - Delete a contact
 - Get a contact
 - Get all contacts
 - Update a contact
- Deal
 - Create a deal
 - Create a new record, or update the current one if it already exists (upsert)
 - Delete a contact
 - Get a contact
 - Get all contacts
 - Update a contact
- Invoice
 - Create an invoice
 - Create a new record, or update the current one if it already exists (upsert)
 - Delete an invoice
 - Get an invoice
 - Get all invoices
 - Update an invoice
- Lead
 - Create a lead
 - Create a new record, or update the current one if it already exists (upsert)
 - Delete a lead
 - Get a lead
 - Get all leads
 - Get lead fields
 - Update a lead
- Product
 - Create a product
 - Create a new record, or update the current one if it already exists (upsert)
 - Delete a product
 - Get a product
 - Get all products
 - Update a product
- Purchase Order
 - Create a purchase order
 - Create a new record, or update the current one if it already exists (upsert)
 - Delete a purchase order
 - Get a purchase order
 - Get all purchase orders
 - Update a purchase order
- Quote
 - Create a quote
 - Create a new record, or update the current one if it already exists (upsert)
 - Delete a quote
 - Get a quote
 - Get all quotes
 - Update a quote
- Sales Order

- Create a sales order
- Create a new record, or update the current one if it already exists (upsert)
- Delete a sales order
- Get a sales order
- Get all sales orders
- Update a sales order
- Vendor
 - Create a vendor
 - Create a new record, or update the current one if it already exists (upsert)
 - Delete a vendor
 - Get a vendor
 - Get all vendors
 - Update a vendor

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Zoom node

Use the Zoom node to automate work in Zoom, and integrate Zoom with other applications. n8n has built-in support for a wide range of Zoom features, including creating, retrieving, deleting, and updating meetings.

On this page, you’ll find a list of operations the Zoom node supports and links to more resources.

-8<- “_snippets/integrations/builtin/app-nodes/ai-tools.md”

Operations

- Meeting
 - Create a meeting
 - Delete a meeting
 - Retrieve a meeting
 - Retrieve all meetings
 - Update a meeting

Templates and examples

-8<- “_snippets/integrations/builtin/app-nodes/operation-not-supported.md”

Zulip node

Use the Zulip node to automate work in Zulip, and integrate Zulip with other applications. n8n has built-in support for a wide range of Zulip features, including creating, deleting, and getting users and streams, as well as sending messages.

On this page, you'll find a list of operations the Zulip node supports and links to more resources.

Operations

- Message
 - Delete a message
 - Get a message
 - Send a private message
 - Send a message to stream
 - Update a message
 - Upload a file
- Stream
 - Create a stream.
 - Delete a stream.
 - Get all streams.
 - Get subscribed streams.
 - Update a stream.
- User
 - Create a user.
 - Deactivate a user.
 - Get a user.
 - Get all users.
 - Update a user.

Templates and examples

Triggers library

This section provides information about [n8n's Triggers](#).

ActiveCampaign Trigger node

[ActiveCampaign](#) is a cloud software platform for small-to-mid-sized business. The company offers software for customer experience automation, which combines the email marketing, marketing automation, sales automation, and CRM categories.

Events

- New ActiveCampaign event

Related resources

n8n provides an app node for ActiveCampaign. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [ActiveCampaign's documentation](#) for details about their API.

Acuity Scheduling Trigger node

[Acuity Scheduling](#) is a cloud-based appointment scheduling software solution that enables business owners to manage their appointments online. It has the capability to automatically sync calendars according to users' time zones and can send regular alerts and reminders to users regarding their appointment schedules.

Events

- Appointment canceled
 - Appointment changed
 - Appointment rescheduled
 - Appointment scheduled
 - Order completed
-

Affinity Trigger node

[Affinity](#) is a powerful relationship intelligence platform enabling teams to leverage their network to close the next big deal.

Events

- Field value
 - Created
 - Deleted
 - Updated
- Field
 - Created
 - Deleted
 - Updated
- File
 - Created
 - Deleted
- List entry
 - Created
 - Deleted
- List
 - Created
 - Deleted
 - Updated
- Note
 - Created
 - Deleted
 - Updated

- Opportunity
 - Created
 - Deleted
 - Updated
- Organization
 - Created
 - Deleted
 - Updated
- Person
 - Created
 - Deleted
 - Updated

Related resources

n8n provides an app node for Affinity. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [Affinity's documentation](#) for details about their API.

Airtable Trigger node

[Airtable](#) is a spreadsheet-database hybrid, with the features of a database but applied to a spreadsheet. The fields in an Airtable table are similar to cells in a spreadsheet, but have types such as 'checkbox', 'phone number', and 'drop-down list', and can reference file attachments like images.

On this page, you'll find a list of events the Airtable Trigger node can respond to and links to more resources.

Events

- **New Airtable event**

Related resources

n8n provides an app node for Airtable. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [Airtable's documentation](#) for details about their API.

Node parameters

Use these parameters to configure your node.

Poll Times

n8n's Airtable node uses polling for check for updates on configured Airtable resources. The **Poll Times** parameter configures the querying frequency:

- Every Minute
- Every Hour
- Every Day
- Every Week
- Every Month
- Every X: Check for updates every given number of minutes or hours.
- Custom: Customize the polling interval by providing a [cron expression](#).

Use the **Add Poll Time** button to add more polling intervals.

Base

The [Airtable base](#) you want to check for updates on. You can provide your base's URL or [base ID](#).

Table

The [Airtable table](#) within the Airtable base that you want to check for updates on. You can provide the table's URL or [table ID](#).

Trigger Field

A created or last modified field in your table. The Airtable Trigger node uses this to determine what updates occurred since the previous check.

Download Attachments

Whether to download attachments from the table. When enabled, the **Download Fields** parameter defines the attachment fields.

Download Fields

When you enable the **Download Attachments** toggle, this field defines which table fields to download. Field names are case sensitive. Use a comma to separate multiple field names.

Additional Fields

Use the **Add Field** button to add the following parameters:

- **Fields:** A comma-separated list of fields to include in the output. If you don't specify anything here, the output will contain only the **Trigger Field**.
- **Formula:** An [Airtable formula](#) to further filter the results. You can use this to add further constraints to the events that trigger the workflow. Note that formula values aren't taken into account for manual executions, only for production polling.
- **View ID:** The name or ID of a table view. When defined, only returns records available in the given view.

AMQP Trigger node

AMQP is an open standard application layer protocol for message-oriented middleware. The defining features of AMQP are message orientation, queuing, routing, reliability and security. This node supports AMQP 1.0 compatible message brokers.

Asana Trigger node

Asana is a web and mobile application designed to help teams organize, track, and manage their work.

Events

- New Asana event

Related resources

n8n provides an app node for Asana. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [Asana's documentation](#) for details about their API.

Autopilot Trigger node

Autopilot is a visual marketing software that allows you to automate and personalize your marketing across the entire customer journey.

Events

- Contact added
 - Contact added to a list
 - Contact entered to a segment
 - Contact left a segment
 - Contact removed from a list
 - Contact unsubscribed
 - Contact updated
-

AWS SNS Trigger node

AWS SNS is a notification service provided as part of Amazon Web Services. It provides a low-cost infrastructure for the mass delivery of messages, predominantly to mobile users.

Events

- New AWS SNS event

Related resources

n8n provides an app node for AWS SNS. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [AWS SNS's documentation](#) for details about their API.

Bitbucket Trigger node

[Bitbucket](#) is a web-based version control repository hosting service owned by Atlassian, for source code and development projects that use either Mercurial or Git revision control systems.

Box Trigger node

[Box](#) is a cloud computing company which provides file sharing, collaborating, and other tools for working with files uploaded to its servers.

Find your Box Target ID

To get your Target ID in Box:

1. Open the file/folder that you would like to monitor.
 2. Copy the string of characters after `folder/` in your URL. This is the target ID. For example, if the URL is `https://app.box.com/folder/12345`, then 12345 is the target ID.
 3. Paste it in the **Target ID** field in n8n.
-

Brevo Trigger node

[Brevo](#) is a digital marketing platform to help users grow their business.

Events

- Email blocked
- Email clicked
- Email deferred
- Email delivered
- Email hard bounce
- Email invalid
- Email marked spam
- Email opened

- Email sent
- Email soft bounce
- Email unique open
- Email unsubscribed

Related resources

n8n provides an app node for Brevo. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [Brevo's documentation](#) for details about their API.

Calendly Trigger node

Calendly is an automated scheduling software that's designed to help find meeting times.

Events

- Event created
- Event canceled

Cal Trigger node

Cal is the event-juggling scheduler for everyone. Focus on meeting, not making meetings.

Events

- Booking cancelled
- Booking created
- Booking rescheduled
- Meeting ended

Chargebee Trigger node

Chargebee is a billing platform for subscription based SaaS and eCommerce businesses. Chargebee integrates with payment gateways to let you automate recurring payment collection along with invoicing, taxes, accounting, email notifications, SaaS Metrics and customer management.

Add webhook URL in Chargebee

To add a Webhook URL in Chargebee:

1. Open your Chargebee dashboard.
 2. Go to **Settings > Configure Chargebee**.
 3. Scroll down and select **Webhooks**.
 4. Select the **Add Webhook** button.
 5. Enter the **Webhook Name** and the **Webhook URL**.
 6. Select **Create**.
-

ClickUp Trigger node

[ClickUp](#) is a cloud-based collaboration and project management tool suitable for businesses of all sizes and industries. Features include communication and collaboration tools, task assignments and statuses, alerts and a task toolbar.

Events

- Key result
 - Created
 - Deleted
 - Updated
- List
 - Created
 - Deleted
 - Updated
- Space
 - Created
 - Deleted
 - Updated
- Task
 - Assignee updated
 - Comment
 - Posted
 - Updated
 - Created
 - Deleted
 - Due date updated
 - Moved
 - Status updated
 - Tag updated
 - Time estimate updated
 - Time tracked updated
 - Updated

Related resources

n8n provides an app node for ClickUp. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [ClickUp's documentation](#) for details about their API.

Clockify Trigger node

[Clockify](#) is a free time tracker and timesheet app for tracking work hours across projects.

This node uses the workflow timezone setting to specify the range of time entries starting time. Configure the timezone in your [Workflow Settings](#) if you want this trigger node to retrieve the right time entries.

ConvertKit Trigger node

[ConvertKit](#) is a fully featured email marketing platform. Use ConvertKit to build an email list, send email broadcasts, automate sequences, create segments, and build landing pages.

Events

- Form subscribe
- Link click
- Product purchase
- Purchase created
- Purchase complete
- Sequence complete
- Sequence subscribe
- Subscriber activated
- Subscriber unsubscribe
- Tag add
- Tag Remove

Related resources

n8n provides an app node for ConvertKit. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [ConvertKit's documentation](#) for details about their API.

Copper Trigger node

[Copper](#) is a CRM that focuses on strong integration with Google Workspace. It's mainly targeted towards small and medium-sized businesses.

Events

- Delete
- New
- Update

Related resources

n8n provides an app node for Copper. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [Copper's documentation](#) for details about their API.

crowd.dev Trigger node

Use the crowd.dev Trigger node to respond to events in [crowd.dev](#) and integrate crowd.dev with other applications. n8n has built-in support for a wide range of crowd.dev events, including new activities and new members.

On this page, you'll find a list of events the crowd.dev Trigger node can respond to and links to more resources.

Events

- New Activity
- New Member

Related resources

n8n provides an app node for crowd.dev. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [crowd.dev's documentation](#) for more information about the service.

Customer.io Trigger node

[Customer.io](#) enables users to send newsletters to selected segments of customers using their website data. You can send targeted emails, push notifications, and SMS to lower churn, create stronger relationships, and drive subscriptions.

Events

- Customer
 - Subscribed
 - Unsubscribe
- Email
 - Bounced
 - Clicked
 - Converted
 - Delivered
 - Drafted

- Failed
- Opened
- Sent
- Spammed
- Push
 - Attempted
 - Bounced
 - Clicked
 - Delivered
 - Drafted
 - Failed
 - Opened
 - Sent
- Slack
 - Attempted
 - Clicked
 - Drafted
 - Failed
 - Sent
- Sms
 - Attempted
 - Bounced
 - Clicked
 - Delivered
 - Drafted
 - Failed
 - Sent

Related resources

n8n provides an app node for Customer.io. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [Customer.io's documentation](#) for details about their API.

Emelia Trigger node

[Emelia](#) is a cold-mailing tool.

Events

- Email Bounced
 - Email Opened
 - Email Replied
 - Email Sent
 - Link Clicked
 - Unsubscribed Contact
-

Eventbrite Trigger node

[Eventbrite](#) is an event management and ticketing website. The service allows users to browse, create, and promote local events.

Facebook Lead Ads Trigger node

Use the Facebook Lead Ads Trigger node to respond to events in [Facebook Lead Ads](#) and integrate Facebook Lead Ads with other applications. n8n has built-in support for responding to new leads.

On this page, you'll find a list of events the Facebook Lead Ads Trigger node can respond to, and links to more resources.

Events

- New lead

Related resources

View [example workflows and related content](#) on n8n's website.

Refer to [Facebook Lead Ads' documentation](#) for details about their API.

Common issues

Here are some common errors and issues with the Facebook Lead Ads Trigger node and steps to resolve or troubleshoot them.

Workflow only works in testing or production

Facebook Lead Ads only allows you to register a single webhook per app. This means that every time you switch from using the testing URL to the production URL (and vice versa), Facebook Lead Ads overwrites the registered webhook URL.

You may have trouble with this if you try to test a workflow that's also active in production. Facebook Lead Ads will only send events to one of the two webhook URLs, so the other will never receive event notifications.

To work around this, you can disable your workflow when testing:

1. Go to your workflow page.
 2. Toggle the **Active** switch in the top panel to disable the workflow temporarily.
 3. Test your workflow using the test webhook URL.
 4. When you finish testing, toggle the **Inactive** toggle to enable the workflow again. The production webhook URL should resume working.
-

Facebook Trigger node

[Facebook](#) is a social networking site to connect and share with family and friends online.

Use the Facebook Trigger node to trigger a workflow when events occur in Facebook.

Objects

- [**Ad Account**](#): Get updates for certain ads changes.
- [**Application**](#): Get updates sent to the application.
- [**Certificate Transparency**](#): Get updates when new security certificates are generated for your subscribed domains, including new certificates and potential phishing attempts.
- Activity and events in a [**Group**](#)
- [**Instagram**](#): Get updates when someone comments on the Media objects of your app users; @mentions your app users; or when Stories of your app users expire.
- [**Link**](#): Get updates about the links for rich previews by an external provider
- [**Page**](#) updates
- [**Permissions**](#): Updates when granting or revoking permissions
- [**User**](#) profile updates
- [**WhatsApp Business Account**](#)
- [**Workplace Security**](#)

For each **Object**, use the **Field Names or IDs** dropdown to select more details on what data to receive. Refer to the linked pages for more details.

Related resources

View [example workflows and related content](#) on n8n's website.

Refer to Meta's [Graph API documentation](#) for details about their API.

Facebook Trigger Ad Account object

Use this object to receive updates on certain ads changes in an Ad Account. Refer to [Facebook Trigger](#) for more information on the trigger itself.

Trigger configuration

To configure the trigger with this Object:

1. Select the **Credential to connect with**. Select an existing or create a new [Facebook App credential](#).
2. Enter the **APP ID** of the app connected to your credential. Refer to the [Facebook App credential](#) documentation for more information.
3. Select **Ad Account** as the **Object**.
4. **Field Names or IDs**: By default, the node will trigger on all the available Ad Account events using the * wildcard filter. If you'd like to limit the events, use the X to remove the star and use the dropdown or an expression to select the updates you're interested in. Options include:
 - **In Process Ad Objects**: Notifies you when a campaign, ad set, or ad exits the IN_PROCESS status. Refer to Meta's [Post-processing for Ad Creation and Edits](#) for more information.
 - **With Issues Ad Objects**: Notifies you when a campaign, ad set, or ad under the ad account receives the WITH_ISSUES status.
5. In **Options**, turn on the toggle to **Include Values**. This Object type fails without the option enabled.

Related resources

Refer to [Webhooks for Ad Accounts](#) and Meta's [Ad Account Graph API](#) reference for more information.

Facebook Trigger Application object

Use this object to receive updates sent to a specific app. Refer to [Facebook Trigger](#) for more information on the trigger itself.

Trigger configuration

To configure the trigger with this Object:

1. Select the **Credential to connect with**. Select an existing or create a new [Facebook App credential](#).
2. Enter the **APP ID** of the app connected to your credential. Refer to the [Facebook App credential](#) documentation for more information.
3. Select **Application** as the **Object**.
4. **Field Names or IDs**: By default, the node will trigger on all the available events using the * wildcard filter. If you'd like to limit the events, use the X to remove the star and use the dropdown or an expression to select the updates you're interested in. Options include:
 - **Add Account**
 - **Ads Rules Engine**
 - **Async Requests**
 - **Async Sessions**
 - **Group Install**
 - **Oe Reseller Onboarding Request Created**
 - **Plugin Comment**
 - **Plugin Comment Reply**

5. In **Options**, turn on the toggle to **Include Values**. This Object type fails without the option enabled.

Related resources

Refer to Meta's [Application Graph API reference](#) for more information.

Facebook Trigger Certificate Transparency object

Use this object to receive updates about newly issued certificates for any domains that you have subscribed for certificate alerts or phishing alerts. Refer to [Facebook Trigger](#) for more information on the trigger itself.

Trigger configuration

To configure the trigger with this Object:

1. Select the **Credential to connect with**. Select an existing or create a new [Facebook App credential](#).
2. Enter the **APP ID** of the app connected to your credential. Refer to the [Facebook App credential documentation](#) for more information.
3. Select **Certificate Transparency** as the **Object**.
4. **Field Names or IDs**: By default, the node will trigger on all the available events using the * wildcard filter. If you'd like to limit the events, use the X to remove the star and use the dropdown or an expression to select the updates you're interested in. Options include:
 - **Certificate**: Notifies you when someone issues a new certificate for your subscribed domains. You'll need to subscribe your domain for certificate alerts.
 - **Phishing**: Notifies you when someone issues a new certificate that may be phishing one of your legitimate subscribed domains.
5. In **Options**, turn on the toggle to **Include Values**. This Object type fails without the option enabled.

For these alerts, you'll need to subscribe your domain to the relevant alerts:

- Refer to [Certificate Alerts](#) for Certificate Alerts subscriptions.
- Refer to [Phishing Alerts](#) for Phishing Alerts subscriptions.

Related resources

Refer to [Webhooks for Certificate Transparency](#) and Meta's [Certificate Transparency Graph API reference](#) for more information.

Facebook Trigger Group object

Use this object to receive updates about activities and events in a group. Refer to [Facebook Trigger](#) for more information on the trigger itself.

Trigger configuration

To configure the trigger with this Object:

1. Select the **Credential to connect with**. Select an existing or create a new [Facebook App credential](#).
2. Enter the **APP ID** of the app connected to your credential. Refer to the [Facebook App credential](#) documentation for more information.
3. Select **Group** as the **Object**.
4. **Field Names or IDs**: By default, the node will trigger on all the available events using the * wildcard filter. If you'd like to limit the events, use the X to remove the star and use the dropdown or an expression to select the updates you're interested in.
5. In **Options**, turn on the toggle to **Include Values**. This Object type fails without the option enabled.

Related resources

Refer to Meta's [Groups Workplace API](#) reference for more information.

Facebook Trigger Instagram object

Use this object to receive updates when someone comments on the Media objects of your app users; @mentions your app users; or when Stories of your app users expire. Refer to [Facebook Trigger](#) for more information on the trigger itself.

Trigger configuration

To configure the trigger with this Object:

1. Select the **Credential to connect with**. Select an existing or create a new [Facebook App credential](#).
2. Enter the **APP ID** of the app connected to your credential. Refer to the [Facebook App credential](#) documentation for more information.
3. Select **Instagram** as the **Object**.
4. **Field Names or IDs**: By default, the node will trigger on all the available events using the * wildcard filter. If you'd like to limit the events, use the X to remove the star and use the dropdown or an expression to select the updates you're interested in. Options include:
 - **Comments**: Notifies you when anyone comments on an IG Media owned by your app's Instagram user.
 - **Messaging Handover**
 - **Mentions**: Notifies you whenever an Instagram user @mentions an Instagram Business or Creator Account in a comment or caption.
 - **Messages**: Notifies you when anyone messages your app's

Instagram user.

- **Messaging Seen:** Notifies you when someone sees a message sent by your app's Instagram user.
 - **Standby**
 - **Story Insights:** Notifies you one hour after a story expires with metrics describing interactions on a story.
5. In **Options**, turn on the toggle to **Include Values**. This Object type fails without the option enabled.

Related resources

Refer to [Webhooks for Instagram](#) and Meta's [Instagram Graph API](#) reference for more information.

Facebook Trigger Link object

Use this object to receive updates about links for rich previews by an external provider. Refer to [Facebook Trigger](#) for more information on the trigger itself.

Trigger configuration

To configure the trigger with this Object:

1. Select the **Credential to connect with**. Select an existing or create a new [Facebook App credential](#).
2. Enter the **APP ID** of the app connected to your credential. Refer to the [Facebook App credential](#) documentation for more information.
3. Select **Link** as the **Object**.
4. **Field Names or IDs:** By default, the node will trigger on all the available events using the * wildcard filter. If you'd like to limit the events, use the X to remove the star and use the dropdown or an expression to select the updates you're interested in.
5. In **Options**, turn on the toggle to **Include Values**. This Object type fails without the option enabled.

Related resources

Refer to Meta's [Links Workplace API](#) reference for more information.

Facebook Trigger Page object

Use this object to receive updates when updates to your page profile fields or profile settings occur or someone mentions your page. Refer to [Facebook Trigger](#) for more information on the trigger itself.

Prerequisites

This Object requires some configuration in your app and page before you can use the trigger:

1. At least one page admin needs to grant the `manage_pages` permission to your app.
2. The page admin needs to have at least moderator privileges. If they don't, they won't receive all content.
3. You'll also need to add the app to your page, and you may need to go to the [Graph API explorer](#) and execute this call with your app token:

```
{page-id}/subscribed_apps?subscribed_fields=feed
```

Trigger configuration

To configure the trigger with this Object:

1. Select the **Credential to connect with**. Select an existing or create a new [Facebook App credential](#).
2. Enter the **APP ID** of the app connected to your credential. Refer to the [Facebook App credential](#) documentation for more information.
3. Select **Page** as the **Object**.
4. **Field Names or IDs**: By default, the node will trigger on all the available events using the * wildcard filter. If you'd like to limit the events, use the X to remove the star and use the dropdown or an expression to select the updates you're interested in. Options include individual profile fields, as well as:
 - **Feed**: Describes most changes to a page's feed, including posts, likes, shares, and so on.
 - **Leadgen**: Notifies you when a page's lead generation settings change.
 - **Live Videos**: Notifies you when a page's live video status changes.
 - **Mention**: Notifies you when new mentions in pages, comments, and so on occur.
 - **Merchant Review**: Notifies you when a page's merchant review settings change.
 - **Page Change Proposal**: Notifies you when Facebook suggests proposed changes for your Facebook Page.
 - **Page Upcoming Change**: Notifies you about upcoming changes that will occur on your Facebook Page. Facebook has suggested these changes and they may have a deadline to accept or reject before automatically taking effect.
 - **Product Review**: Notifies you when a page's product review settings change.
 - **Ratings**: Notifies you when a page's ratings change, including new ratings or when a user comments on or reacts to a rating.
 - **Videos**: Notifies you when the encoding status of a video on a page changes.
5. In **Options**, turn on the toggle to **Include Values**. This Object type fails without the option enabled.

Related resources

Refer to [Webhooks for Pages](#) and Meta's [Page Graph API](#) reference for more information.

Facebook Trigger Permissions object

Use this object to receive updates when a user grants or revokes a permission for your app. Refer to [Facebook Trigger](#) for more information on the trigger itself.

Trigger configuration

To configure the trigger with this Object:

1. Select the **Credential to connect with**. Select an existing or create a new [Facebook App credential](#).
2. Enter the **APP ID** of the app connected to your credential. Refer to the [Facebook App credential](#) documentation for more information.
3. Select **Permissions** as the **Object**.
4. **Field Names or IDs**: By default, the node will trigger on all the available events using the * wildcard filter. If you'd like to limit the events, use the X to remove the star and use the dropdown or an expression to select the updates you're interested in.
5. In **Options**, choose whether to turn on the toggle to **Include Values**. When turned on, the node includes the new values for the changes.

Related resources

Refer to Meta's [Permissions Graph API](#) reference for more information.

Facebook Trigger User object

Use this object to receive updates when changes to a user's profile occur. Refer to [Facebook Trigger](#) for more information on the trigger itself.

Trigger configuration

To configure the trigger with this Object:

1. Select the **Credential to connect with**. Select an existing or create a new [Facebook App credential](#).
2. Enter the **APP ID** of the app connected to your credential. Refer to the [Facebook App credential](#) documentation for more information.
3. Select **User** as the **Object**.
4. **Field Names or IDs**: By default, the node will trigger on all the available events using the * wildcard filter. If you'd like to limit the events, use the X to remove the star and use the dropdown or an expression to select the updates you're interested in.
5. In **Options**, choose whether to turn on the toggle to **Include**

Values. When turned on, the node includes the new values for the changes.

Related resources

Refer to Meta's [User Graph API](#) reference for more information.

Facebook Trigger WhatsApp Business Account object

Use this object to receive updates when your WhatsApp Business Account (WABA) changes. Refer to [Facebook Trigger](#) for more information on the trigger itself.

Prerequisites

This Object requires some configuration in your app and WhatsApp account before you can use the trigger:

1. Subscribe your app under your WhatsApp business account. You must subscribe an app owned by your business. Apps shared with your business can't receive webhook notifications.
2. If you are working as a Solution Partner, make sure your app has completed App Review and requested the `whatsapp_business_management` permission.

Trigger configuration

To configure the trigger with this Object:

1. Select the **Credential to connect with**. Select an existing or create a new [Facebook App credential](#).
2. Enter the **APP ID** of the app connected to your credential. Refer to the [Facebook App credential](#) documentation for more information.
3. Select **WhatsApp Business Account** as the **Object**.
4. **Field Names or IDs:** By default, the node will trigger on all the available events using the * wildcard filter. If you'd like to limit the events, use the X to remove the star and use the dropdown or an expression to select the updates you're interested in. Options include:
 - **Message Template Status Update**
 - **Phone Number Name Update**
 - **Phone Number Quality Update**
 - **Account Review Update**
 - **Account Update**
5. In **Options**, turn on the toggle to **Include Values**. This Object type fails without the option enabled.

Refer to [Webhooks for WhatsApp Business Accounts](#) and Meta's [WhatsApp Business Account Graph API](#) reference for more information.

Facebook Trigger Workplace Security object

Use this object to receive updates when Workplace security events occur, like adding or removing admins, users joining or leaving a Workplace, and more. Refer to [Facebook Trigger](#) for more information on the trigger itself.

Trigger configuration

To configure the trigger with this Object:

1. Select the **Credential to connect with**. Select an existing or create a new [Facebook App credential](#).
2. Enter the **APP ID** of the app connected to your credential. Refer to the [Facebook App credential](#) documentation for more information.
3. Select **Workplace Security** as the **Object**.
4. **Field Names or IDs**: By default, the node will trigger on all the available events using the * wildcard filter. If you'd like to limit the events, use the X to remove the star and use the dropdown or an expression to select the updates you're interested in.
5. In **Options**, turn on the toggle to **Include Values**. This Object type fails without the option enabled.

Related resources

Refer to Meta's [Security Workplace API](#) reference for more information.

Figma Trigger (Beta) node

[Figma](#) is a prototyping tool which is primarily web-based, with more offline features enabled by desktop applications for macOS and Windows.

Events

- **File Commented**: Triggers when someone comments on a file.
 - **File Deleted**: Triggers when someone deletes an individual file, but not when someone deletes an entire folder with all files.
 - **File Updated**: Triggers when someone saves or deletes a file. A save occurs when someone closes a file within 30 seconds after making changes.
 - **File Version Updated**: Triggers when someone creates a named version in the version history of a file.
 - **Library Publish**: Triggers when someone publishes a library file.
-

Flow Trigger node

[Flow](#) is modern task and project management software for teams. It brings together tasks, projects, timelines, and conversations, and integrates with a lot of tools.

Events

- New Flow event

Related resources

n8n provides an app node for Flow. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [Flow's documentation](#) for details about their API.

Form.io Trigger node

[Form.io](#) is an enterprise class combined form and API data management platform for building complex form-based business process applications.

Formstack Trigger node

[Formstack](#) is a workplace productivity platform that helps organizations streamline digital work through no-code online forms, documents, and signatures.

GetResponse Trigger node

[GetResponse](#) is an online platform that offers email marketing software, landing page creator, webinar hosting, and much more.

Events

- Receive notifications when a customer is subscribed to a list
 - Receive notifications when a customer is unsubscribed from a list
 - Receive notifications when an email is opened
 - Receive notifications when an email is clicked
 - Receive notifications when a survey is submitted
-

GitHub Trigger node

[GitHub](#) provides hosting for software development and version control using Git. It offers the distributed version control and source code management (SCM) functionality of Git, access control and several collaboration features such as bug tracking, feature requests, task management, and wikis for every project.

Events

- Check run
- Check suite
- Commit comment
- Create
- Delete
- Deploy key
- Deployment
- Deployment status
- Fork
- GitHub app authorization
- Gollum
- Installation
- Installation repositories
- Issue comment
- Label
- Marketplace purchase
- Member
- Membership
- Meta
- Milestone
- Org block
- Organization
- Page build
- Project
- Project card
- Project column
- Public
- Pull request
- Pull request review
- Pull request review comment
- Push
- Release
- Repository
- Repository import
- Repository vulnerability alert
- Security advisory
- Star
- Status
- Team
- Team add
- Watch

Related resources

n8n provides an app node for GitHub. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [GitHub's documentation](#) for details about their API.

GitLab Trigger node

[GitLab](#) is a web-based DevOps lifecycle tool that provides a Git-repository manager providing wiki, issue-tracking, and continuous integration/continuous installation pipeline features.

Events

- Comment
- Confidential issues
- Confidential comments
- Deployments
- Issue
- Job
- Merge request
- Pipeline
- Push
- Release
- Tag
- Wiki page

Related resources

n8n provides an app node for GitLab. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [GitLab's documentation](#) for details about their API.

Gmail Trigger node

[Gmail](#) is an email service developed by Google. The Gmail Trigger node can start a workflow based on events in Gmail.

Events

- **Message Received:** The node triggers for new messages at the selected **Poll Time**.

Node parameters

Configure the node with these parameters:

- **Credential to connect with:** Select or create a new Google credential to use for the trigger. Refer to [Google credentials](#) for more information on setting up a new credential.
- **Poll Times:** Select a poll **Mode** to set how often to trigger the poll. Your **Mode** selection will add or remove relevant fields. Refer to [Poll Mode options](#) to configure the parameters for each mode

type.

- **Simplify:** Choose whether to return a simplified version of the response (turned on, default) or the raw data (turned off).
 - The simplified version returns email message IDs, labels, and email headers, including: From, To, CC, BCC, and Subject.

Node filters

Use these filters to further refine the node's behavior:

- **Include Spam and Trash:** Select whether the node should trigger on new messages in the Spam and Trash folders (turned on) or not (turned off).
- **Label Names or IDs:** Only trigger on messages with the selected labels added to them. Select the Label names you want to apply or enter an expression to specify IDs. The dropdown populates based on the **Credential** you selected.
- **Search:** Enter Gmail search refine filters, like `from:`, to trigger the node on the filtered conditions only. Refer to [Refine searches in Gmail](#) for more information.
- **Read Status:** Choose whether to receive **Unread and read emails, Unread emails only** (default), or **Read emails only**.
- **Sender:** Enter an email or a part of a sender name to trigger only on messages from that sender.

Related resources

n8n provides an app node for Gmail. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [Google's Gmail API documentation](#) for details about their API.

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

Gmail Trigger node Poll Mode options

Use the [Gmail Trigger node's Poll Time](#) parameter to set how often to trigger the poll. Your **Mode** selection will add or remove relevant fields.

Poll mode options

Refer to the sections below for details on using each **Mode**.

-8<- “_snippets/integrations/builtin/poll-modes.md”

Gmail Trigger node common issues

Here are some common errors and issues with the [Gmail Trigger node](#) and steps to resolve or troubleshoot them.

401 unauthorized error

The full text of the error looks like this:

```
401 - {"error": "unauthorized_client", "error_description": "Client is unauthorized to retrieve access tokens using this method, or client not authorized for any of the scopes requested."}
```

This error occurs when there's an issue with the credential you're using and its scopes or permissions.

To resolve:

1. For [OAuth2](#) credentials, make sure you've enabled the Gmail API in [APIs & Services > Library](#). Refer to [Google OAuth2 Single Service - Enable APIs](#) for more information.
 2. For [Service Account](#) credentials:
 1. [Enable domain-wide delegation](#).
 2. Make sure you add the Gmail API as part of the domain-wide delegation configuration.
-

Google Calendar Trigger node

[Google Calendar](#) is a time-management and scheduling calendar service developed by Google.

Events

- [Event Cancelled](#)
- [Event Created](#)
- [Event Ended](#)
- [Event Started](#)
- [Event Updated](#)

Related resources

n8n provides an app node for Google Calendar. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [Google Calendar's documentation](#) for details about their API.

Google Drive Trigger node

[Google Drive](#) is a file storage and synchronization service developed by Google. It allows users to store files on their servers, synchronize files across devices, and share files.

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

Google Drive Trigger node common issues

Here are some common errors and issues with the [Google Drive Trigger node](#) and steps to resolve or troubleshoot them.

401 unauthorized error

The full text of the error looks like this:

```
401 - {"error": "unauthorized_client", "error_description": "Client is unauthorized to retrieve access tokens using this method, or client not authorized for any of the scopes requested."}
```

This error occurs when there's an issue with the credential you're using and its scopes or permissions.

To resolve:

1. For [OAuth2](#) credentials, make sure you've enabled the Google Drive API in **APIs & Services > Library**. Refer to [Google OAuth2 Single Service - Enable APIs](#) for more information.
2. For [Service Account](#) credentials:
 1. [Enable domain-wide delegation](#).
 2. Make sure you add the Google Drive API as part of the domain-wide delegation configuration.

Handling more than one file change

The Google Drive Trigger node polls Google Drive for changes at a set interval (once every minute by default).

If multiple changes to the **Watch For** criteria occur during the polling interval, a single Google Drive Trigger event occurs containing the changes as items. To handle this, your workflow must account for times when the data might contain more than one item.

You can use an [if node](#) or a [switch node](#) to change your workflow's behavior depending on whether the data from the Google Drive Trigger node contains a single item or multiple items.

Google Business Profile Trigger

node

Use the Google Business Profile Trigger node to respond to events in [Google Business Profile](#) and integrate Google Business Profile with other applications. n8n has built-in support for responding to new reviews.

On this page, you'll find a list of events the Google Business Profile Trigger node can respond to and links to more resources.

Events

- Review Added

Related resources

n8n provides an app node for Google Business Profile. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [Google Business Profile's documentation](#) for details about their API.

Google Sheets Trigger node

[Google Sheets](#) is a web-based spreadsheet program that's part of Google's office software suite within its Google Drive service.

Events

- Row added
- Row updated
- Row added or updated

Related resources

Refer to [Google Sheet's API documentation](#) for more information about the service.

n8n provides an app node for Google Sheets. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

Google Sheets Trigger node common issues

Here are some common errors and issues with the [Google Sheets Trigger node](#) and steps to resolve or troubleshoot them.

Stuck waiting for trigger event

When testing the Google Sheets Trigger node with the **Execute step** or **Execute workflow** buttons, the execution may appear stuck and unable to stop listening for events. If this occurs, you may need to exit the workflow and open it again to reset the canvas.

Stuck listening events often occur due to issues with your network configuration outside of n8n. Specifically, this behavior often occurs when you run n8n behind a reverse proxy without configuring websocket proxying.

To resolve this issue, check your reverse proxy configuration (Nginx, Caddy, Apache HTTP Server, Traefik, etc.) to enable websocket support.

Date and time columns are rendering as numbers

Google Sheets can render dates and times a few different ways.

The **serial number** format, popularized by Lotus 1-2-3 and used my types of spreadsheet software, represents dates as a decimal number. The whole number component (the part left of the decimal) represents the number of days since December 30, 1899. The decimal portion (the part right of the decimal) represents time as a portion of a 24-hour period (for example, .5 represents noon).

To use a different format for date and time values, adjust the format in your Google Sheet Trigger node. This is available when **Trigger On** is set to **Row Added**:

1. Open the Google Sheet Trigger node on your canvas.
2. Select **Add option**.
3. Select **DateTime Render**.
4. Change **DateTime Render** to **Formatted String**.

The Google Sheets Trigger node will now format date, time, datetime, and duration fields as strings according to their number format.

The number format depends on the spreadsheet's locale settings. You can change the local by opening the spreadsheet and selecting **File > Settings**. In the **General** tab, set **Locale** to your preferred locale. Select **Save settings** to adjust the value.

Gumroad Trigger node

[Gumroad](#) is an online platform that enables creators to sell products directly to consumers.

Help Scout Trigger node

[Help Scout](#) is a help desk software that provides an email-based customer support platform, knowledge base tool, and an embeddable search/contact widget for customer service professionals.

HubSpot Trigger node

[HubSpot](#) provides tools for social media marketing, content management, web analytics, landing pages, customer support, and search engine optimization.

Events

- Company
 - Created
 - Deleted
 - Property changed
- Contact
 - Created
 - Deleted
 - Privacy deleted
 - Property changed
- Conversation
 - Created
 - Deleted
 - New message
 - Privacy deletion
 - Property changed
- Deal
 - Created
 - Deleted
 - Property changed
- Ticket
 - Created
 - Deleted
 - Property changed

Related resources

n8n provides an app node for HubSpot. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [HubSpot's documentation](#) for details about their API.

Invoice Ninja Trigger node

[Invoice Ninja](#) is a free open-source online invoicing app for freelancers & businesses. It offers invoicing, payments, expense tracking, & time-tasks.

Jira Trigger node

[Jira](#) is a proprietary issue tracking product developed by Atlassian that allows bug tracking and agile project management.

JotForm Trigger node

[JotForm](#) is an online form building service. JotForm's software creates forms with a drag and drop creation tool and an option to encrypt user data.

Kafka Trigger node

[Kafka](#) is an open-source distributed event streaming platform that one can use for high-performance data pipelines, streaming analytics, data integration, and mission-critical applications.

Keap Trigger node

[Keap](#) is an e-mail marketing and sales platform for small businesses, including products to manage and optimize the customer lifecycle, customer relationship management, marketing automation, lead capture, and e-commerce.

KoboToolbox Trigger node

[KoboToolbox](#) is a field survey and data collection tool to design interactive forms to be completed offline from mobile devices. It's available both as a free cloud solution or as a self-hosted version.

This node starts a workflow upon new submissions of a specified form. The trigger node handles the creation/deletion of the hook, so you don't need to do any setup in KoboToolbox.

It works the same way as the Get Submission operation in the [KoboToolbox](#) node, including supporting the same reformatting options.

Lemlist Trigger node

Lemlist is an email outreach platform that allows you to automatically generate personalized images and videos and send personalized cold emails.

Events

- °
 - Aircall Created
 - Aircall Done
 - Aircall Ended
 - Aircall Interested
 - Aircall Not Interested
 - Api Done
 - Api Failed
 - Api Interested
 - Api Not Interested
 - Attracted
 - Connection Issue
 - Contacted
 - Custom Domain Errors
 - Emails Bounced
 - Emails Clicked
 - Emails Failed
 - Emails Interested
 - Emails Not Interested
 - Emails Opened
 - Emails Replied
 - Emails Send Failed
 - Emails Sent
 - Emails Unsubscribed
 - Hooked
 - Interested
 - Lemwarm Paused
 - LinkedIn Interested
 - LinkedIn Invite Accepted
 - LinkedIn Invite Done
 - LinkedIn Invite Failed
 - LinkedIn Not Interested
 - LinkedIn Replied
 - LinkedIn Send Failed
 - LinkedIn Sent
 - LinkedIn Visit Done
 - LinkedIn Visit Failed
 - LinkedIn Voice Note Done
 - LinkedIn Voice Note Failed
 - Manual Interested
 - Manual Not Interested
 - Not Interested
 - Opportunities Done
 - Paused
 - Resumed
 - Send Limit Reached
 - Skipped
 - Warmed
-

Linear Trigger node

[Linear](#) is a SaaS issue tracking tool.

Events

- Comment Reaction
 - Cycle
 - Issue
 - Issue Comment
 - Issue Label
 - Project
-

LoneScale Trigger node

Use the LoneScale Trigger node to respond to workflow events in [LoneScale](#) and integrate LoneScale with other applications.

On this page, you'll find a list of operations the LoneScale node supports, and links to more resources.

Events

- On new LoneScale event

Related resources

n8n provides an app node for LoneScale. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Mailchimp Trigger node

[Mailchimp](#) is an integrated marketing platform that allows business owners to automate their email campaigns and track user engagement.

MailerLite Trigger node

[MailerLite](#) is an email marketing solution that provides you with a user-friendly content editor, simplified subscriber management, and campaign reports with the most important statistics.

On this page, you'll find a list of events the MailerLite Trigger node can respond to and links to more resources.

Events

- Campaign Sent
 - Subscriber Added to Group
 - Subscriber Automation Completed
 - Subscriber Automation Triggered
 - Subscriber Bounced
 - Subscriber Created
 - Subscriber Complained
 - Subscriber Removed from Group
 - Subscriber Unsubscribe
 - Subscriber Updated
-

Mailjet Trigger node

[Mailjet](#) is a cloud-based email sending and tracking system. The platform allows professionals to send both marketing emails and transactional emails. It includes tools for designing emails, sending massive volumes and tracking these messages.

Mautic Trigger node

[Mautic](#) is an open-source marketing automation software that helps online businesses automate their repetitive marketing tasks such as lead generation, contact scoring, contact segmentation, and marketing campaigns.

Related resources

n8n provides an app node for Mautic. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Microsoft OneDrive Trigger node

Use the Microsoft OneDrive Trigger node to respond to events in [Microsoft OneDrive](#) and integrate Microsoft OneDrive with other applications. n8n has built-in support for file and folder events in OneDrive.

On this page, you'll find a list of events the Microsoft OneDrive Trigger node can respond to and links to more resources.

Events

- On File Created
- On File Updated
- On Folder Created
- On Folder Updates

Related resources

n8n provides an app node for Microsoft OneDrive. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [Microsoft's OneDrive API documentation](#) for more information about the service.

Microsoft Outlook Trigger node

Use the Microsoft Outlook Trigger node to respond to events in [Microsoft Outlook](#) and integrate Microsoft Outlook with other applications.

On this page, you'll find a list of events the Microsoft Outlook Trigger node can respond to, and links to more resources.

Events

- Message Received

Related resources

n8n provides an app node for Microsoft Outlook. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Microsoft Teams Trigger node

Use the Microsoft Teams Trigger node to respond to events in [Microsoft Teams](#) and integrate Microsoft Teams with other applications.

On this page, you'll find a list of events the Microsoft Teams Trigger node can respond to and links to more resources.

Events

- **New Channel**
- **New Channel Message**
- **New Chat**
- **New Chat Message**
- **New Team Member**

Related resources

n8n provides an app node for Microsoft Teams. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to the [Microsoft Teams documentation](#) for details about their API.

MQTT Trigger node

[MQTT](#) is an open OASIS and ISO standard lightweight, publish-subscribe network protocol that transports messages between devices.

Netlify Trigger node

[Netlify](#) offers hosting and serverless backend services for web applications and static websites.

Related resources

n8n provides an app node for Netlify. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Notion Trigger node

[Notion](#) is an all-in-one workspace for your notes, tasks, wikis, and databases.

Events

- Page added to database
- Page updated in database

Related resources

n8n provides an app node for Notion. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [Notion's documentation](#) for details about their API.

Onfleet Trigger node

[Onfleet](#) is a logistics platform offering a last-mile delivery solution.

Events

Trigger a workflow on:

- SMS recipient opt out
 - SMS recipient response missed
 - Task arrival
 - Task assigned
 - Task cloned
 - Task completed
 - Task created
 - Task delayed
 - Task ETA
 - Task failed
 - Task started
 - Task unassigned
 - Task updated
 - Worker created
 - Worker deleted
 - Worker duty
-

PayPal Trigger node

[PayPal](#) is a digital payment service that supports online fund transfers that customers can use when shopping online.

Pipedrive Trigger node

[Pipedrive](#) is a cloud-based sales software company that aims to improve the productivity of businesses through the use of their software.

Postgres Trigger node

Use the Postgres Trigger node to respond to events in [Postgres](#) and integrate Postgres with other applications. n8n has built-in support responding to insert, update, and delete events.

Events

You can configure how the node listens for events.

- Select **Listen and Create Trigger Rule**, then choose the events to listen for:
 - Insert
 - Update
 - Delete
- Select **Listen to Channel**, then enter a channel name that the node should monitor.

Related resources

n8n provides an app node for Postgres. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Postmark Trigger node

[Postmark](#) helps deliver and track application email. You can track statistics such as the number of emails sent or processed, opens, bounces and, spam complaints.

Pushcut Trigger node

[Pushcut](#) is an app for iOS that lets you create smart notifications to kick off shortcuts, URLs, and online automation.

Configure a Pushcut action

Follow these steps to configure your Pushcut Trigger node with your Pushcut app.

1. In your Pushcut app, select a notification from the **Notifications** screen.
 2. Select the **Add Action** button.
 3. Enter an action name in the **Label** field.
 4. Select the **Server** tab.
 5. Select the **Integration** tab.
 6. Select **Integration Trigger**.
 7. In n8n, enter a name for the action and select **Execute step**.
 8. Select this action under the **Select Integration Trigger** screen in your Pushcut app.
 9. Select **Done** in the top right to save the action.
-

RabbitMQ Trigger node

[RabbitMQ](#) is an open-source message broker that accepts and forwards messages.

Related resources

n8n provides an app node for RabbitMQ. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Redis Trigger node

Redis is an open-source, in-memory data structure store, used as a database, cache and message broker.

Use the Redis Trigger node to subscribe to a Redis channel. The workflow starts whenever the channel receives a new message.

Salesforce Trigger node

Use the Salesforce Trigger node to respond to events in [Salesforce](#) and integrate Salesforce with other applications. n8n has built-in support for a wide range of Salesforce events.

On this page, you'll find a list of events the Salesforce Trigger node can respond to, and links to more resources.

Events

- On Account Created
- On Account Updated
- On Attachment Created
- On Attachment Updated
- On Case Created
- On Case Updated
- On Contact Created
- On Contact Updated
- On Custom Object Created
- On Custom Object Updated
- On Lead Created
- On Lead Updated
- On Opportunity Created
- On Opportunity Updated
- On Task Created
- On Task Updated
- On User Created
- On User Updated

Related resources

n8n provides an app node for Salesforce. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

SeaTable Trigger node

SeaTable is a collaborative database application with a spreadsheet interface.

Shopify Trigger node

Shopify is an e-commerce platform that allows users to set up an online store and sell their products.

Slack Trigger node

Use the Slack Trigger node to respond to events in [Slack](#) and integrate Slack with other applications. n8n has built-in support for a wide range of Slack events, including new messages, reactions, and new channels.

On this page, you'll find a list of events the Slack Trigger node can respond to and links to more resources.

Events

- **Any Event:** The node triggers on any event in Slack.
- **Bot / App Mention:** The node triggers when your bot or app is [mentioned](#) in a channel the app is in.
- **File Made Public:** The node triggers when a file is [made public](#).
- **File Shared:** The node triggers when a file is [shared](#) in a channel the app is in.
- **New Message Posted to Channel:** The node triggers when a new message is posted to a channel the app is in.
- **New Public Channel Created:** The node triggers when a new [public channel](#) is created.
- **New User:** The node triggers when a new user is added to Slack.
- **Reaction Added:** The node triggers when a [reaction](#) is added to a message the app is added to.

Parameters

Once you've set the events to trigger on, use the remaining parameters to further define the node's behavior:

- **Watch Whole Workspace:** Whether the node should watch for the selected **Events** in all channels in the workspace (turned on) or not (turned off, default).
- **Channel to Watch:** Select the channel your node should watch for the selected **Events**. This parameter only appears if you don't turn on **Watch Whole Workspace**. You can select a channel:
 - **From list:** The node uses your credential to look up a list of channels in the workspace so you can select the channel you want.
 - **By ID:** Enter the ID of a channel you want to watch. Slack displays the channel ID at the bottom of the channel details with a one-click copy button.
 - **By URL:** Enter the URL of the channel you want to watch, formatted as <https://app.slack.com/client/<channel-address>>.

- **Download Files:** Whether to download files and use them in the node's output (turned on) or not (turned off, default). Use this parameter with the **File Made Public** and **File Shared** events.

Options

You can further refine the node's behavior when you **Add Options**:

- **Resolve IDs:** Whether to resolve the IDs to their respective names and return them (turned on) or not (turned off, default).
- **Usernames or IDs to ignore:** Select usernames or enter a comma-separated string of encoded user IDs to ignore events from. Choose from the list, or specify IDs using an [expression](#).

Related resources

n8n provides an app node for Slack. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [Slack's documentation](#) for details about their API.

Required scopes

To use this node, you need to create an application in Slack and enable event subscriptions. Refer to [Slack credentials | Slack Trigger configuration](#) for more information.

You must add the appropriate scopes to your Slack app for this trigger node to work.

The node requires scopes for the `conversations.list` and `users.list` methods at minimum. Check out the [Scopes | Slack credentials](#) list for a more complete list of scopes.

Verify the webhook

From version 1.106.0, you can set a [Slack Signing Secret](#) when configuring your [Slack credentials](#). When set, the Slack trigger node automatically verifies that requests are from Slack and include a trusted signature. n8n recommends setting this to ensure you only process requests sent from Slack.

Common issues

Here are some common errors and issues with the Slack Trigger node and steps to resolve or troubleshoot them.

Workflow only works in testing or production

Slack only allows you to register a single webhook per app. This means that you can't switch from using the testing URL to the production URL (and vice versa) without reconfiguring the registered webhook URL.

You may have trouble with this if you try to test a workflow that's also active in production. Slack will only send events to one of the two webhook URLs, so the other will never receive event notifications.

To work around this, you can disable your workflow when testing:

1. Go to your workflow page.
2. Toggle the **Active** switch in the top panel to disable the workflow temporarily.
3. Edit the **Request URL** in your the [Slack Trigger configuration](#) to use the testing webhook URL instead of the production webhook URL.
4. Test your workflow using the test webhook URL.
5. When you finish testing, edit the **Request URL** in your the [Slack Trigger configuration](#) to use the production webhook URL instead of the testing webhook URL.
6. Toggle the **Inactive** toggle to enable the workflow again. The production webhook URL should resume working.

Token expired

-8<- “_snippets/integrations/builtin/credentials/slack/token-rotation.md”

Strava Trigger node

Strava is an internet service for tracking human exercise which incorporates social network features.

Events

- **[All]**
 - [All]
 - Created
 - Deleted
 - Updated
- **Activity**
 - [All]
 - Created
 - Deleted
 - Updated
- **Athlete**
 - [All]
 - Created
 - Deleted
 - Updated

Stripe Trigger node

Stripe is a suite of payment APIs that powers commerce for online businesses.

SurveyMonkey Trigger node

[SurveyMonkey](#) is an online cloud-based SaaS survey platform that also provides a suite of paid back-end programs.

Taiga Trigger node

[Taiga](#) is a free and open-source project management platform for startups, agile developers, and designers.

Telegram Trigger node

[Telegram](#) is a cloud-based instant messaging and voice over IP service. Users can send messages and exchange photos, videos, stickers, audio, and files of any type. On this page, you'll find a list of events the Telegram Trigger node can respond to and links to more resources.

Events

- *: All updates except "Chat Member", "Message Reaction", and "Message Reaction Count" (default behavior of Telegram API as they produce a lot of calls of updates).
- **Business Connection:** Trigger when the bot is connected to or disconnected from a business account, or a user edited an existing connection with the bot.
- **Business Message:** Trigger on a new message from a connected business account.
- **Callback Query:** Trigger on new incoming callback query.
- **Channel Post:** Trigger on new incoming channel post of any kind — including text, photo, sticker, and so on.
- **Chat Boost:** Trigger when a chat boost is added or changed. The bot must be an administrator in the chat to receive these updates.
- **Chat Join Request:** Trigger when a request to join the chat is sent. The bot must have the `can_invite_users` administrator right in the chat to receive these updates.
- **Chat Member:** Trigger when a chat member's status is updated. The bot must be an administrator in the chat.
- **Chosen Inline Result:** Trigger when the result of an inline query chosen by a user is sent. Please see Telegram's API documentation on [feedback collection](#) for details on how to enable these updates for your bot.
- **Deleted Business Messages:** Trigger when messages are deleted from a connected business account.
- **Edited Business Message:** Trigger on new version of a message from a connected business account.
- **Edited Channel Post:** Trigger on new version of a channel post that is known to the bot is edited.
- **Edited Message:** Trigger on new version of a channel post that is known to the bot is edited.
- **Inline Query:** Trigger on new incoming inline query.
- **Message:** Trigger on new incoming message of any kind — text,

photo, sticker, and so on.

- **Message Reaction:** Trigger when a reaction to a message is changed by a user. The bot must be an administrator in the chat. The update isn't received for reactions set by bots.
- **Message Reaction Count:** Trigger when reactions to a message with anonymous reactions are changed. The bot must be an administrator in the chat. The updates are grouped and can be sent with delay up to a few minutes.
- **My Chat Member:** Trigger when the bot's chat member status is updated in a chat. For private chats, this update is received only when the bot is blocked or unblocked by the user.
- **Poll:** Trigger on new poll state. Bots only receive updates about stopped polls and polls which are sent by the bot.
- **Poll Answer:** Trigger when user changes their answer in a non-anonymous poll. Bots only receive new votes in polls that were sent by the bot itself.
- **Pre-Checkout Query:** Trigger on new incoming pre-checkout query. Contains full information about checkout.
- **Purchased Paid Media:** Trigger when a user purchases paid media with a non-empty payload sent by the bot in a non-channel chat.
- **Removed Chat Boost:** Trigger when a boost is removed from a chat. The bot must be an administrator in the chat to receive these updates.
- **Shipping Query:** Trigger on new incoming shipping query. Only for invoices with flexible price.

Some **events may require additional permissions**, see [Telegram's API documentation](#) for more information.

Options

- **Download Images/Files:** Whether to download attached images or files to include in the output data.
 - **Image Size:** When you enable **Download Images/Files**, this configures the size of image to download. Downloads large images by default.
- **Restrict to Chat IDs:** Only trigger for events with the listed chat IDs. You can include multiple chat IDs separated by commas.
- **Restrict to User IDs:** Only trigger for events with the listed user IDs. You can include multiple user IDs separated by commas.

Related resources

n8n provides an app node for Telegram. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [Telegram's API documentation](#) for details about their API.

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

Telegram Trigger node common issues

Here are some common errors and issues with the [Telegram Trigger node](#) and steps to resolve or troubleshoot them.

Stuck waiting for trigger event

When testing the Telegram Trigger node with the **Execute step** or **Execute workflow** buttons, the execution may appear stuck and unable to stop listening for events. If this occurs, you may need to exit the workflow and open it again to reset the canvas.

Stuck listening events often occur due to issues with your network configuration outside of n8n. Specifically, this behavior often occurs when you run n8n behind a reverse proxy without configuring websocket proxying.

To resolve this issue, check your reverse proxy configuration (Nginx, Caddy, Apache HTTP Server, Traefik, etc.) to enable websocket support.

Bad request: bad webhook: An HTTPS URL must be provided for webhook

This error occurs when you run n8n behind a reverse proxy and there is a problem with your instance's webhook URL.

When running n8n behind a reverse proxy, you must [configure the WEBHOOK_URL environment variable](#) with the public URL where your n8n instance is running. For Telegram, this URL must use HTTPS.

To fix this issue, configure TLS/SSL termination in your reverse proxy. Afterward, update your `WEBHOOK_URL` environment variable to use the HTTPS address.

Workflow only works in testing or production

Telegram only allows you to register a single webhook per app. This means that every time you switch from using the testing URL to the production URL (and vice versa), Telegram overwrites the registered webhook URL.

You may have trouble with this if you try to test a workflow that's also active in production. The Telegram bot will only send events to one of the two webhook URLs, so the other will never receive event notifications.

To work around this, you can either disable your workflow when testing or create separate Telegram bots for testing and production.

To create a separate telegram bot for testing, repeat the process you completed to create your first bot. Reference [Telegram's bot documentation](#) and the [Telegram bot API reference](#) for more

information.

To disable your workflow when testing, try the following:

1. Go to your workflow page.
 2. Toggle the **Active** switch in the top panel to disable the workflow temporarily.
 3. Test your workflow using the test webhook URL.
 4. When you finish testing, toggle the **Inactive** toggle to enable the workflow again. The production webhook URL should resume working.
-

TheHive 5 Trigger node

Use the TheHive 5 Trigger node to respond to events in [TheHive](#) and integrate TheHive with other applications. n8n has built-in support for a wide range of TheHive events, including alerts, cases, comments, pages, and tasks.

On this page, you'll find a list of events the TheHive5 Trigger node can respond to and links to more resources.

Events

- Alert
 - Created
 - Deleted
 - Updated
- Case
 - Created
 - Deleted
 - Updated
- Comment
 - Created
 - Deleted
 - Updated
- Observable
 - Created
 - Deleted
 - Updated
- Page
 - Created
 - Deleted
 - Updated
- Task
 - Created
 - Deleted
 - Updated
- Task log
 - Created
 - Deleted
 - Updated

Related resources

n8n provides an app node for TheHive 5. You can find the node docs [here](#).

Refer to TheHive's [documentation](#) for more information about the service.

Configure a webhook in TheHive

To configure the webhook for your TheHive instance:

1. Copy the testing and production webhook URLs from TheHive Trigger node.
2. Add the following lines to the application.conf file. This is TheHive configuration file:

```
notification.webhook.endpoints = [
  {
    name: TESTING_WEBHOOK_NAME
    url: TESTING_WEBHOOK_URL
    version: 1
    wsConfig: {}
    includedTheHiveOrganisations: ["ORGANIZATION_NAME"]
    excludedTheHiveOrganisations: []
  },
  {
    name: PRODUCTION_WEBHOOK_NAME
    url: PRODUCTION_WEBHOOK_URL
    version: 1
    wsConfig: {}
    includedTheHiveOrganisations: ["ORGANIZATION_NAME"]
    excludedTheHiveOrganisations: []
  }
]
```

3. Replace TESTING_WEBHOOK_URL and PRODUCTION_WEBHOOK_URL with the URLs you copied in the previous step.
4. Replace TESTING_WEBHOOK_NAME and PRODUCTION_WEBHOOK_NAME with your preferred endpoint names.
5. Replace ORGANIZATION_NAME with your organization name.
6. Execute the following cURL command to enable notifications: sh
curl -XPUT -uTHEHIVE_USERNAME:THEHIVE_PASSWORD -H 'Content-type: application/json'
THEHIVE_URL/api/config/organisation/notification -d ' {
 "value": [{ "trigger": { "name": "AnyEvent" }, "notifier": { "name": "webhook", "endpoint": "TESTING_WEBHOOK_NAME" } }, { "trigger": { "name": "AnyEvent" }, "notifier": { "name": "webhook", "endpoint": "PRODUCTION_WEBHOOK_NAME" } }] }'

TheHive Trigger node

On this page, you'll find a list of events the TheHive Trigger node can respond to and links to more resources.

Events

- Alert
 - Created
 - Deleted
 - Updated
- Case
 - Created
 - Deleted
 - Updated
- Log
 - Created
 - Deleted
 - Updated
- Observable
 - Created
 - Deleted
 - Updated
- Task
 - Created
 - Deleted
 - Updated

Related resources

n8n provides an app node for TheHive. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to TheHive's documentation for more information about the service:

- [Version 3](#)
- [Version 4](#)

Configure a webhook in TheHive

To configure the webhook for your TheHive instance:

1. Copy the testing and production webhook URLs from TheHive Trigger node.
2. Add the following lines to the application.conf file. This is TheHive configuration file:

```
notification.webhook.endpoints = [
  {
    name: TESTING_WEBHOOK_NAME
    url: TESTING_WEBHOOK_URL
    version: 0
    wsConfig: {}
    includedTheHiveOrganisations: ["ORGANIZATION_NAME"]
    excludedTheHiveOrganisations: []
  }
]
```

```

},
{
  name: PRODUCTION_WEBHOOK_NAME
  url: PRODUCTION_WEBHOOK_URL
  version: 0
  wsConfig: {}
  includedTheHiveOrganisations: ["ORGANIZATION_NAME"]
  excludedTheHiveOrganisations: []
}
]

```

3. Replace TESTING_WEBHOOK_URL and PRODUCTION_WEBHOOK_URL with the URLs you copied in the previous step.
4. Replace TESTING_WEBHOOK_NAME and PRODUCTION_WEBHOOK_NAME with your preferred endpoint names.
5. Replace ORGANIZATION_NAME with your organization name.
6. Execute the following cURL command to enable notifications: sh

```
curl -XPUT -uTHEHIVE_USERNAME:THEHIVE_PASSWORD -H 'Content-type: application/json'
THEHIVE_URL/api/config/organisation/notification -d '{
  "value": [
    {
      "trigger": {
        "name": "AnyEvent"
      },
      "notifier": {
        "name": "webhook",
        "endpoint": "TESTING_WEBHOOK_NAME"
      }
    },
    {
      "trigger": {
        "name": "AnyEvent"
      },
      "notifier": {
        "name": "webhook",
        "endpoint": "PRODUCTION_WEBHOOK_NAME"
      }
    }
  ]
}'
```

Toggl Trigger node

[Toggl](#) is a time tracking app that offers online time tracking and reporting services through their website along with mobile and desktop applications.

Trello Trigger node

[Trello](#) is a web-based Kanban-style list-making application which is a subsidiary of Atlassian. Users can create their task boards with different columns and move the tasks between them.

Find the Model ID

The model ID is the ID of any model in Trello. Depending on the use-case, it could be the User ID, List ID, and so on.

For this specific example, the List ID would be the Model ID:

1. Open the Trello board that contains the list.
2. If the list doesn't have any cards, add a card to the list.
3. Open the card, add .json at the end of the URL, and press enter.
4. In the JSON file, you will see a field called idList.

5. Copy idList and paste it in the **Model ID** field in n8n.
-

Twilio Trigger node

Use the Twilio Trigger node to respond to events in [Twilio](#) and integrate Twilio with other applications. n8n has built-in support for a wide range of Twilio events, including new SMS and calls.

On this page, you'll find a list of events the Twilio Trigger node can respond to and links to more resources.

Events

- On New SMS
- On New Call

Related resources

n8n provides an app node for Twilio. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [Twilio's documentation](#) for details about their API.

Typeform Trigger node

[Typeform](#) is an online software as a service company that specializes in online form building and online surveys. Its main software creates dynamic forms based on user needs.

Venafi TLS Protect Cloud Trigger node

[Venafi](#) is a cybersecurity company providing services for machine identity management. They offer solutions to manage and protect identities for a wide range of machine types, delivering global visibility, lifecycle automation, and actionable intelligence.

Use the n8n Venafi TLS Protect Cloud Trigger node to start a workflow in n8n in response to events in the [cloud-based Venafi TLS Protect](#) service.

Webex by Cisco Trigger node

[Webex by Cisco](#) is a web conferencing and videoconferencing application.

Webflow Trigger node

[Webflow](#) is an application that allows you to build responsive websites with browser-based visual editing software.

WhatsApp Trigger node

Use the WhatsApp Trigger node to respond to events in WhatsApp and integrate WhatsApp with other applications. n8n has built-in support for a wide range of WhatsApp events, including account, message, and phone number events.

On this page, you'll find a list of events the WhatsApp Trigger node can respond to, and links to more resources.

Events

- Account Review Update
- Account Update
- Business Capability Update
- Message Template Quality Update
- Message Template Status Update
- Messages
- Phone Number Name Update
- Phone Number Quality Update
- Security
- Template Category Update

Related resources

n8n provides an app node for WhatsApp. You can find the node docs [here](#).

View [example workflows and related content](#) on n8n's website.

Refer to [WhatsApp's documentation](#) for details about their API.

Common issues

Here are some common errors and issues with the WhatsApp Trigger node and steps to resolve or troubleshoot them.

Workflow only works in testing or production

WhatsApp only allows you to register a single webhook per app. This means that every time you switch from using the testing URL to the production URL (and vice versa), WhatsApp overwrites the registered webhook URL.

You may have trouble with this if you try to test a workflow that's also active in production. WhatsApp will only send events to one of the two webhook URLs, so the other will never receive event notifications.

To work around this, you can disable your workflow when testing:

1. Go to your workflow page.
 2. Toggle the **Active** switch in the top panel to disable the workflow temporarily.
 3. Test your workflow using the test webhook URL.
 4. When you finish testing, toggle the **Inactive** toggle to enable the workflow again. The production webhook URL should resume working.
-

Wise Trigger node

[Wise](#) allows you to transfer money abroad with low-cost money transfers, receive money with international account details, and track transactions on your phone.

Events

- Triggered every time a balance account is credited
 - Triggered every time a balance account is credited or debited
 - Triggered every time a transfer's list of active cases is updated
 - Triggered every time a transfer's status is updated
-

WooCommerce Trigger node

[WooCommerce](#) is a customizable, open-source e-commerce plugin for WordPress.

Events

- coupon.created
 - coupon.updated
 - coupon.deleted
 - customer.created
 - customer.updated
 - customer.deleted
 - order.created
 - order.updated
 - order.deleted
 - product.created
 - product.updated
 - product.deleted
-

Workable Trigger node

Use the Workable Trigger node to respond to events in the [Workable](#) recruiting platform and integrate Workable with other applications. n8n has built-in support for a wide range of Workable events, including candidate created and moved.

On this page, you'll find a list of events the Workable Trigger node can respond to and links to more resources.

Events

- [Candidate Created](#)
- [Candidate Moved](#)

Related resources

View [example workflows and related content](#) on n8n's website.

Refer to [Workable's API documentation](#) for details about using the service.

Wufoo Trigger node

[Wufoo](#) is an online form builder that helps you create custom HTML forms without writing code.

Zendesk Trigger node

[Zendesk](#) is a support ticketing system, designed to help track, prioritize, and solve customer support interactions. More than just a help desk, Zendesk Support helps nurture customer relationships with personalized, responsive support across any channel.

Cluster nodes

-8<- “_snippets/integrations/builtin/cluster-nodes/cluster-nodes-summary.md”

Root nodes

Each cluster starts with one [root node](#).

Sub-nodes

Each root node can have one or more [sub-nodes](#) attached to it.

Root nodes

Root nodes are the foundational nodes within a group of cluster nodes.

-8<- “_snippets/integrations/builtin/cluster-nodes/cluster-nodes-summary.md”

AI Agent node

An [AI agent](#) is an autonomous system that receives data, makes rational decisions, and acts within its environment to achieve specific goals. The AI agent's environment is everything the agent can access that isn't the agent itself. This agent uses external [tools](#) and APIs to perform actions and retrieve information. It can understand the capabilities of different tools and determine which tool to use depending on the task.

Templates and examples

Related resources

Refer to [LangChain’s documentation on agents](#) for more information about the service.

New to AI Agents? Read the [n8n blog introduction to AI agents](#).

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Common issues

For common errors or issues and suggested resolution steps, refer to [Common Issues](#).

Conversational AI Agent node

The Conversational Agent has human-like conversations. It can maintain context, understand user intent, and provide relevant answers. This agent is typically used for building chatbots, virtual assistants, and customer support systems.

The Conversational Agent describes [tools](#) in the system prompt and parses JSON responses for tool calls. If your preferred AI model doesn't support tool calling or you're handling simpler interactions, this agent is a good general option. It's more flexible but may be less accurate than the [Tools Agent](#).

Refer to [AI Agent](#) for more information on the AI Agent node itself.

```
-8<- "snippets/integrations/builtin/cluster-nodes/use-with-chat-trigger.md"
```

Node parameters

Configure the Conversational Agent using the following parameters.

Prompt

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/prompt.md"
```

Require Specific Output Format

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/output-format.md"
```

Node options

Refine the Conversational Agent node's behavior using these options:

Human Message

Tell the agent about the tools it can use and add context to the user's input.

You must include these expressions and variable:

- `{tools}`: A LangChain expression that provides a string of the tools you've connected to the Agent. Provide some context or explanation about who should use the tools and how they should use them.
- `{format_instructions}`: A LangChain expression that provides the schema or format from the output parser node you've connected. Since the instructions themselves are context, you don't need to provide context for this expression.
- `{{input}}`: A LangChain variable containing the user's prompt. This variable populates with the value of the **Prompt** parameter. Provide some context that this is the user's input.

Here's an example of how you might use these strings:

Example:

```
TOOLS
-----
Assistant can ask the user to use tools to look up information that
may be helpful in answering the user's original question. The tools
the human can use are:

{tools}

{format_instructions}

USER'S INPUT
-----
Here is the user's input (remember to respond with a markdown code
```

```
snippet of a JSON blob with a single action, and NOTHING else):  
{{input}}
```

System Message

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/system-message.md"
```

Max Iterations

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/max-iterations.md"
```

Return Intermediate Steps

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/return-intermediate-steps.md"
```

Templates and examples

Refer to the main AI Agent node's [Templates and examples](#) section.

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

OpenAI Functions Agent node

Use the OpenAI Functions Agent node to use an [OpenAI functions model](#). These are models that detect when a function should be called and respond with the inputs that should be passed to the function.

Refer to [AI Agent](#) for more information on the AI Agent node itself.

```
-8<- "_snippets/integrations/builtin/cluster-nodes/use-with-chat-trigger.md"
```

Node parameters

Configure the OpenAI Functions Agent using the following parameters.

Prompt

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/prompt.md"
```

Require Specific Output Format

```
-8<- " snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/output-format.md"
```

Node options

Refine the OpenAI Functions Agent node's behavior using these options:

System Message

```
-8<- " snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/system-message.md"
```

Max Iterations

```
-8<- " snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/max-iterations.md"
```

Return Intermediate Steps

```
-8<- " snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/return-intermediate-steps.md"
```

Templates and examples

Refer to the main AI Agent node's [Templates and examples](#) section.

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

Plan and Execute Agent node

The Plan and Execute Agent is like the [ReAct agent](#) but with a focus on planning. It first creates a high-level plan to solve the given task and then executes the plan step by step. This agent is most useful for tasks that require a structured approach and careful planning.

Refer to [AI Agent](#) for more information on the AI Agent node itself.

Node parameters

Configure the Plan and Execute Agent using the following parameters.

Prompt

```
-8<- " snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/prompt.md"
```

Require Specific Output Format

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/output-format.md”

Node options

Refine the Plan and Execute Agent node’s behavior using these options:

Human Message Template

Enter a message that n8n will send to the agent during each step execution.

Available LangChain expressions:

- {previous_steps}: Contains information about the previous steps the agent’s already completed.
- {current_step}: Contains information about the current step.
- {agent_scratchpad}: Information to remember for the next iteration.

Templates and examples

Refer to the main AI Agent node’s [Templates and examples](#) section.

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

ReAct AI Agent node

The ReAct Agent node implements [ReAct](#) logic. ReAct (reasoning and acting) brings together the reasoning powers of chain-of-thought prompting and action plan generation.

The ReAct Agent reasons about a given task, determines the necessary actions, and then executes them. It follows the cycle of reasoning and acting until it completes the task. The ReAct agent can break down complex tasks into smaller sub-tasks, prioritise them, and execute them one after the other.

Refer to [AI Agent](#) for more information on the AI Agent node itself.

Node parameters

Configure the ReAct Agent using the following parameters.

Prompt

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/prompt.md"
```

Require Specific Output Format

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/output-format.md"
```

Node options

Use the options to create a message to send to the agent at the start of the conversation. The message type depends on the model you're using:

- **Chat models:** These models have the concept of three components interacting (AI, system, and human). They can receive system messages and human messages (prompts).
- **Instruct models:** These models don't have the concept of separate AI, system, and human components. They receive one body of text, the instruct message.

Human Message Template

Use this option to extend the user prompt. This is a way for the agent to pass information from one iteration to the next.

Available LangChain expressions:

- {input}: Contains the user prompt.
- {agent_scratchpad}: Information to remember for the next iteration.

Prefix Message

Enter text to prefix the tools list at the start of the conversation. You don't need to add the list of tools. LangChain automatically adds the tools list.

Suffix Message for Chat Model

Add text to append after the tools list at the start of the conversation when the agent uses a chat model. You don't need to add the list of tools. LangChain automatically adds the tools list.

Suffix Message for Regular Model

Add text to append after the tools list at the start of the conversation when the agent uses a regular/instruct model. You don't need to add the list of tools. LangChain automatically adds the tools list.

Return Intermediate Steps

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/return-intermediate-steps.md"
```

Related resources

Refer to LangChain's [ReAct Agents](#) documentation for more information.

Templates and examples

Refer to the main AI Agent node's [Templates and examples](#) section.

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

SQL AI Agent node

The SQL Agent uses a SQL database as a data source. It can understand natural language questions, convert them into SQL queries, execute the queries, and present the results in a user-friendly format. This agent is valuable for building natural language interfaces to databases.

Refer to [AI Agent](#) for more information on the AI Agent node itself.

Node parameters

Configure the SQL Agent using the following parameters.

Data Source

Choose the database to use as a data source for the node. Options include:

- **MySQL:** Select this option to use a MySQL database.
 - Also select the **Credential for MySQL**.
- **SQLite:** Select this option to use a SQLite database.
 - You must add a [Read/Write File From Disk](#) node before the Agent to read your SQLite file.
 - Also enter the **Input Binary Field** name of your SQLite file coming from the Read/Write File From Disk node.
- **Postgres:** Select this option to use a Postgres database.
 - Also select the **Credential for Postgres**.

Prompt

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/prompt.md"
```

Node options

Refine the SQL Agent node's behavior using these options:

Ignored Tables

If you'd like the node to ignore any tables from the database, enter a comma-separated list of tables you'd like it to ignore.

If left empty, the agent doesn't ignore any tables.

Include Sample Rows

Enter the number of sample rows to include in the prompt to the agent. Default is 3.

Sample rows help the agent understand the schema of the database, but they also increase the number of tokens used.

Included Tables

If you'd only like to include specific tables from the database, enter a comma-separated list of tables to include.

If left empty, the agent includes all tables.

Prefix Prompt

Enter a message you'd like to send to the agent before the **Prompt** text. This initial message can provide more context and guidance to the agent about what it can and can't do, and how to format the response.

n8n fills this field with an example.

Suffix Prompt

Enter a message you'd like to send to the agent after the **Prompt** text.

Available LangChain expressions:

- `{chatHistory}`: A history of messages in this conversation, useful for maintaining context.
- `{input}`: Contains the user prompt.
- `{agent_scratchpad}`: Information to remember for the next iteration.

n8n fills this field with an example.

Limit

Enter the maximum number of results to return.

Default is 10.

Templates and examples

Refer to the main AI Agent node's [Templates and examples](#) section.

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

Tools AI Agent node

The Tools Agent uses external [tools](#) and APIs to perform actions and retrieve information. It can understand the capabilities of different tools and determine which tool to use depending on the task. This agent helps integrate LLMs with various external services and databases.

This agent has an enhanced ability to work with tools and can ensure a standard output format.

The Tools Agent implements [Langchain's tool calling interface](#). This interface describes available tools and their schemas. The agent also has improved output parsing capabilities, as it passes the parser to the model as a formatting tool.

Refer to [AI Agent](#) for more information on the AI Agent node itself.

-8<- “_snippets/integrations/builtin/cluster-nodes/use-with-chat-trigger.md”

This agent supports the following chat models:

- [OpenAI Chat Model](#)
- [Groq Chat Model](#)
- [Mistral Cloud Chat Model](#)
- [Anthropic Chat Model](#)
- [Azure OpenAI Chat Model](#)

??? Details “The Tools Agent can use the following tools...” * [Call n8n Workflow](#) * [Code](#) * [HTTP Request](#) * [Action Network](#) * [ActiveCampaign](#) * [Affinity](#) * [Agile CRM](#) * [Airtable](#) * [APITemplate.io](#) * [Asana](#) * [AWS Lambda](#) * [AWS S3](#) * [AWS SES](#) * [AWS Textract](#) * [AWS Transcribe](#) * [Baserow](#) * [Bubble](#) * [Calculator](#) * [ClickUp](#) * [CoinGecko](#) * [Compression](#) * [Crypto](#) * [DeepL](#) * [DHL](#) * [Discord](#) * [Dropbox](#) * [Elasticsearch](#) * [ERPNext](#) * [Facebook Graph API](#) * [FileMaker](#) * [Ghost](#) * [Git](#) * [GitHub](#) * [GitLab](#) * [Gmail](#) * [Google Analytics](#) * [Google BigQuery](#) * [Google Calendar](#) * [Google Chat](#) * [Google Cloud Firestore](#) * [Google Cloud Realtime Database](#) * [Google Contacts](#) * [Google Docs](#) * [Google Drive](#) * [Google Sheets](#) * [Google Slides](#) * [Google Tasks](#) * [Google Translate](#) * [Google Workspace Admin](#) * [Gotify](#) * [Grafana](#) * [GraphQL](#) * [Hacker News](#) * [Home Assistant](#) * [HubSpot](#) * [Jenkins](#) * [Jira Software](#) * [JWT](#) * [Kafka](#) * [LDAP](#) * [Line](#) * [LinkedIn](#) * [Mailcheck](#) * [Mailgun](#) * [Mattermost](#) * [Mautic](#) * [Medium](#) * [Microsoft Excel 365](#) * [Microsoft OneDrive](#) * [Microsoft Outlook](#) * [Microsoft SQL](#) * [Microsoft Teams](#) * [Microsoft To Do](#) * [Monday.com](#) * [MongoDB](#) * [MQTT](#) * [MySQL](#) * [NASA](#) * [Nextcloud](#) * [NocoDB](#) * [Notion](#) * [Odoo](#) * [OpenWeatherMap](#) * [Pipedrive](#) * [Postgres](#) * [Pushover](#) * [QuickBooks Online](#) * [QuickChart](#) * [RabbitMQ](#) * [Reddit](#) * [Redis](#) * [RocketChat](#) * [S3](#) * [Salesforce](#) * [Send Email](#) * [SendGrid](#) * [SerpApi \(Google Search\)](#) * [Shopify](#) * [Slack](#) * [Spotify](#) * [Stripe](#) * [Supabase](#) * [Telegram](#) * [Todoist](#) * [TOTP](#) * [Trello](#) * [Twilio](#) * [urlscan.io](#) * [Vector Store](#) * [Webflow](#) * [Wikipedia](#) * [WolframAlpha](#) * [WooCommerce](#) * [Wordpress](#) * [X \(Formerly Twitter\)](#) * [YouTube](#) * [Zendesk](#) * [Zoho CRM](#) * [Zoom](#)

Node parameters

Configure the Tools Agent using the following parameters.

Prompt

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/prompt.md"
```

Require Specific Output Format

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/output-format.md"
```

Node options

Refine the Tools Agent node's behavior using these options:

System Message

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/system-message.md"
```

Max Iterations

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/max-iterations.md"
```

Return Intermediate Steps

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/return-intermediate-steps.md"
```

Automatically Passthrough Binary Images

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/binary-images.md"
```

Enable Streaming

When enabled, the AI Agent sends data back to the user in real-time as it generates the answer. This is useful for long-running generations. This is enabled by default.

Templates and examples

Refer to the main AI Agent node's [Templates and examples](#) section.

Dynamic parameters for tools with \$fromAI()

To learn how to dynamically populate parameters for app node tools, refer to [Let AI specify tool parameters with \\$fromAI\(\)](#).

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

AI Agent node common issues

Here are some common errors and issues with the [AI Agent node](#) and steps to resolve or troubleshoot them.

Internal error: 400 Invalid value for 'content'

A full error message might look like this:

```
Internal error
Error: 400 Invalid value for 'content': expected a string, got null.
<stack-trace>
```

This error can occur if the **Prompt** input contains a null value.

You might see this in one of two scenarios:

1. When you've set the **Prompt** to **Define below** and have an expression in your **Text** that isn't generating a value.
 - To resolve, make sure your expressions reference valid fields and that they resolve to valid input rather than null.
2. When you've set the **Prompt** to **Connected Chat Trigger Node** and the incoming data has null values.
 - To resolve, remove any null values from the chatInput field of the input node.

Error in sub-node Simple Memory

This error displays when n8n runs into an issue with the [Simple Memory](#) sub-node.

It most often occurs when your workflow or the workflow template you copied uses an older version of the Simple memory node (previously known as "Window Buffer Memory").

Try removing the Simple Memory node from your workflow and re-adding it, which will guarantee you're using the latest version of the node.

A Chat Model sub-node must be connected error

This error displays when n8n tries to execute the node without having a Chat Model connected.

To resolve this, click the + Chat Model button at the bottom of your screen when the node is open, or click the Chat Model + connector when the node is closed. n8n will then open a selection of possible Chat Models to pick from.

No prompt specified error

This error occurs when the agent expects to get the prompt from the previous node automatically. Typically, this happens when you're using the [Chat Trigger Node](#).

To resolve this issue, find the **Prompt** parameter of the AI Agent node and change it from **Connected Chat Trigger Node** to **Define below**. This allows you to manually build your prompt by referencing output data from other nodes or by adding static text.

Basic LLM Chain node

Use the Basic LLM Chain node to set the prompt that the model will use along with setting an optional parser for the response.

On this page, you'll find the node parameters for the Basic LLM Chain node and links to more resources.

Node parameters

Prompt

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/prompt.md”

Require Specific Output Format

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/output-format.md”

Chat Messages

Use **Chat Messages** when you're using a chat model to set a message.

n8n ignores these options if you don't connect a chat model. Select the **Type Name or ID** you want the node to use:

AI

Enter a sample expected response in the **Message** field. The model will try to respond in the same way in its messages.

System

Enter a system **Message** to include with the user input to help guide the model in what it should do.

Use this option for things like defining tone, for example: Always respond talking like a pirate.

User

Enter a sample user input. Using this with the AI option can help improve the output of the agent. Using both together provides a sample of an input and expected response (the **AI Message**) for the model to follow.

Select one of these input types:

- **Text**: Enter a sample user input as a text **Message**.
- **Image (Binary)**: Select a binary input from a previous node.
Enter the **Image Data Field Name** to identify which binary field from the previous node contains the image data.
- **Image (URL)**: Use this option to feed an image in from a URL.
Enter the **Image URL**.

For both the **Image** types, select the **Image Details** to control how the model processes the image and generates its textual understanding. Choose from:

- **Auto**: The model uses the auto setting, which looks at the image input size and decide if it should use the Low or High setting.
- **Low**: The model receives a low-resolution 512px x 512px version of the image and represents the image with a budget of 65 tokens. This allows the API to return faster responses and consume fewer input tokens. Use this option for use cases that don't require high detail.
- **High**: The model can access the low-resolution image and then creates detailed crops of input images as 512px squares based on the input image size. Each of the detailed crops uses twice the token budget (65 tokens) for a total of 129 tokens. Use this option for use cases that require high detail.

Templates and examples

Related resources

Refer to [LangChain's documentation on Basic LLM Chains](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Common issues

Here are some common errors and issues with the Basic LLM Chain node and steps to resolve or troubleshoot them.

No prompt specified error

This error displays when the **Prompt** is empty or invalid.

You might see this error in one of two scenarios:

1. When you've set the **Prompt** to **Define below** and haven't entered anything in the **Text** field.
 - To resolve, enter a valid prompt in the **Text** field.
 2. When you've set the **Prompt** to **Connected Chat Trigger Node** and the incoming data has no field called `chatInput`.
 - The node expects the `chatInput` field. If your previous node doesn't have this field, add an Edit Fields (Set) node to edit an incoming field name to `chatInput`.
-

Question and Answer Chain node

Use the Question and Answer Chain node to use a vector store as a retriever.

On this page, you'll find the node parameters for the Question and Answer Chain node, and links to more resources.

Node parameters

Query

The question you want to ask.

Templates and examples

Related resources

Refer to LangChain's documentation on retrieval chains for examples of how LangChain can use a vector store as a retriever.

-8<- “ snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Common issues

For common errors or issues and suggested resolution steps, refer to Common Issues.

Question and Answer Chain node common issues

Here are some common errors and issues with the [Question and Answer Chain node](#) and steps to resolve or troubleshoot them.

No prompt specified error

This error displays when the **Prompt** is empty or invalid.

You might see this in one of two scenarios:

1. When you've set the **Prompt** to **Define below** and have an expression in your **Text** that isn't generating a value.
 - To resolve, enter a valid prompt in the **Text** field.
 - Make sure any expressions reference valid fields and that they resolve to valid input rather than null.
2. When you've set the **Prompt** to **Connected Chat Trigger Node** and the incoming data has null values.
 - To resolve, make sure your input contains a `chatInput` field. Add an [Edit Fields \(Set\)](#) node to edit an incoming field name to `chatInput`.
 - Remove any null values from the `chatInput` field of the input node.

A Retriever sub-node must be connected error

This error displays when n8n tries to execute the node without having a Retriever connected.

To resolve this, click the + Retriever button at the bottom of your screen when the node is open, or click the Retriever + connector when the node isn't open. n8n will then open a selection of possible Retrievers to pick from.

Can't produce longer responses

If you need to generate longer responses than the Question and Answer Chain node produces by default, you can try one or more of the following techniques:

- **Connect a more verbose model:** Some AI models produce more terse results than others. Swapping your model for one with a larger context window and more verbose output can increase the word length of your responses.
- **Increase the maximum number of tokens:** Many model nodes (for example the [OpenAI Chat Model](#)) include a **Maximum Number of Tokens** option. You can set this to increase the maximum number of tokens the model can use to produce a response.
- **Build larger responses in stages:** For more detailed answers, you may want to construct replies in stages using a variety of AI nodes. You can use AI split up a single question into multiple prompts and create responses for each. You can then compose a final reply by combining the responses again. Though the details are different, you can find a good example of the general idea in this [template for writing a WordPress post with AI](#).

Summarization Chain node

Use the Summarization Chain node to summarize multiple documents.

On this page, you'll find the node parameters for the Summarization Chain node, and links to more resources.

Node parameters

Choose the type of data you need to summarize in **Data to Summarize**. The data type you choose determines the other node parameters.

- **Use Node Input (JSON) and Use Node Input (Binary)**: summarize the data coming into the node from the workflow.
 - You can configure the **Chunking Strategy**: choose what strategy to use to define the data chunk sizes.
 - If you choose **Simple (Define Below)** you can then set **Characters Per Chunk** and **Chunk Overlap (Characters)**.
 - Choose **Advanced** if you want to connect a splitter sub-node that provides more configuration options.
- **Use Document Loader**: summarize data provided by a document loader sub-node.

Node Options

You can configure the summarization method and prompts. Select **Add Option > Summarization Method and Prompts**.

Options in **Summarization Method**:

- **Map Reduce**: this is the recommended option. Learn more about [Map Reduce](#) in the LangChain documentation.
- **Refine**: learn more about [Refine](#) in the LangChain documentation.
- **Stuff**: learn more about [Stuff](#) in the LangChain documentation.

You can customize the **Individual Summary Prompts** and the **Final Prompt to Combine**. There are examples in the node. You must include the "{text}" placeholder.

Templates and examples

Related resources

Refer to [LangChain's documentation on summarization](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Information Extractor node

Use the Information Extractor node to extract structured information from incoming data.

On this page, you'll find the node parameters for the Information Extractor node, and links to more resources.

Node parameters

- **Text** defines the input text to extract information from. This is usually an expression that references a field from the input items. For example, this could be `{{ $json.chatInput }}` if the input is a chat trigger, or `{{ $json.text }}` if a previous node is Extract from PDF.
- Use **Schema Type** to choose how you want to describe the desired output data format. You can choose between:
 - **From Attribute Descriptions:** This option allows you to define the schema by specifying the list of attributes and their descriptions.
 - **Generate From JSON Example:** Input an example JSON object to automatically generate the schema. The node uses the object property types and names. It ignores the actual values. n8n treats every field as mandatory when generating schemas from JSON examples.
 - **Define using JSON Schema:** Manually input the JSON schema. Read the JSON Schema [guides and examples](#) for help creating a valid JSON schema.

Node options

- **System Prompt Template:** Use this option to change the system prompt that's used for the information extraction. n8n automatically appends format specification instructions to the prompt.

Related resources

-8<- “ snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Text Classifier node

Use the Text Classifier node to classify (categorize) incoming data. Using the categories provided in the parameters (see below), each item is passed to the model to determine its category.

On this page, you'll find the node parameters for the Text Classifier node, and links to more resources.

Node parameters

- **Input Prompt** defines the input to classify. This is usually an expression that references a field from the input items. For example, this could be `{{ $json.chatInput }}` if the input is a chat trigger. By default it references the text field.
- **Categories:** Add the categories that you want to classify your input as. Categories have a name and a description. Use the description to tell the model what the category means. This is important if the meaning isn't obvious. You can add as many categories as you like.

Node options

- **Allow Multiple Classes To Be True:** You can configure the classifier to always output a single class per item (turned off), or allow the model to select multiple classes (turned on).
- **When No Clear Match:** Define what happens if the model can't find a good match for an item. There are two options:
 - **Discard Item** (the default): If the node doesn't detect any of the categories, it drops the item.
 - **Output on Extra, 'Other' Branch:** Creates a separate output branch called **Other**. When the node doesn't detect any of the categories, it outputs items in this branch.
- **System Prompt Template:** Use this option to change the system prompt that's used for the classification. It uses the `{categories}` placeholder for the categories.
- **Enable Auto-Fixing:** When enabled, the node automatically fixes model outputs to ensure they match the expected format. Do this by sending the schema parsing error to the LLM and asking it to fix it.

Related resources

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Sentiment Analysis node

Use the Sentiment Analysis node to analyze the sentiment of incoming text data.

The language model uses the **Sentiment Categories** in the node options to determine each item's sentiment.

Node parameters

- **Text to Analyze** defines the input text for sentiment analysis. This is an expression that references a field from the input items. For example, this could be `{{ $json.chatInput }}` if the input is from a chat or message source. By default, it expects a text field.

Node options

- **Sentiment Categories:** Define the categories that you want to classify your input as.
 - By default, these are Positive, Neutral, Negative. You can customize these categories to fit your specific use case, such as Very Positive, Positive, Neutral, Negative, Very Negative for more granular analysis.
- **Include Detailed Results:** When turned on, this option includes sentiment strength and confidence scores in the output. Note that these scores are estimates generated by the language model and are rough indicators rather than precise measurements.
- **System Prompt Template:** Use this option to change the system prompt that's used for the sentiment analysis. It uses the {categories} placeholder for the categories.
- **Enable Auto-Fixing:** When enabled, the node automatically fixes model outputs to ensure they match the expected format. Do this by sending the schema parsing error to the LLM and asking it to fix it.

Usage Notes

Model Temperature Setting

It's strongly advised to set the temperature of the connected language model to 0 or a value close to 0. This helps ensure that the results are as deterministic as possible, providing more consistent and reliable sentiment analysis across multiple runs.

Language Considerations

The node's performance may vary depending on the language of the input text.

For best results, ensure your chosen language model supports the input language.

Processing Large Volumes

When analyzing large amounts of text, consider splitting the input into smaller chunks to optimize processing time and resource usage.

Iterative Refinement

For complex sentiment analysis tasks, you may need to iteratively refine the system prompt and categories to achieve the desired results.

Example Usage

Basic Sentiment Analysis

1. Connect a data source (for example, RSS Feed, HTTP Request) to the Sentiment Analysis node.

2. Set the “Text to Analyze” field to the relevant item property (for example, `{{ $json.content }}` for blog post content).
3. Keep the default sentiment categories.
4. Connect the node’s outputs to separate paths for processing positive, neutral, and negative sentiments differently.

Custom Category Analysis

1. Change the **Sentiment Categories** to Excited, Happy, Neutral, Disappointed, Angry.
2. Adjust your workflow to handle these five output categories.
3. Use this setup to analyze customer feedback with more nuanced emotional categories.

Related resources

-8<- “[_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md](#)”

LangChain Code node

Use the LangChain Code node to import LangChain. This means if there is functionality you need that n8n hasn’t created a node for, you can still use it. By configuring the LangChain Code node connectors you can use it as a normal node, root node or sub-node.

On this page, you’ll find the node parameters, guidance on configuring the node, and links to more resources.

Node parameters

Add Code

Add your custom code. Choose either **Execute** or **Supply Data** mode. You can only use one mode.

Unlike the [Code node](#), the LangChain Code node doesn’t support Python.

- **Execute:** use the LangChain Code node like n8n’s own Code node. This takes input data from the workflow, processes it, and returns it as the node output. This mode requires a main input and output. You must create these connections in **Inputs** and **Outputs**.
- **Supply Data:** use the LangChain Code node as a sub-node, sending data to a root node. This uses an output other than main.

By default, you can’t load built-in or external modules in this node. Self-hosted users can [enable built-in and external modules](#).

Inputs

Choose the input types.

The main input is the normal connector found in all n8n workflows. If you have a main input and output set in the node, **Execute** code is required.

Outputs

Choose the output types.

The main output is the normal connector found in all n8n workflows. If you have a main input and output set in the node, **Execute** code is required.

Node inputs and outputs configuration

By configuring the LangChain Code node connectors (inputs and outputs) you can use it as an app node, root node or sub-node.

[Image: Screenshot of a workflow with four LangChain nodes, configured as different node types]

| Node type | Inputs | Outputs | Code mode |
|---|-------------------------------|--|-------------|
| App node. Similar to the <u>Code node</u> . | Main | Main | Execute |
| Root node | Main; at least one other type | Main | Execute |
| Sub-node | - | A type other than main.
Must match the input type you want to connect to. | Supply Data |
| Sub-node with sub-nodes | A type other than main | A type other than main.
Must match the input type you want to connect to. | Supply Data |

Built-in methods

n8n provides these methods to make it easier to perform common tasks in the LangChain Code node.

-8<- “ snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/langchaincode/builtin-methods.md”

Templates and examples

Related resources

-8<- “ snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Azure AI Search Vector Store node

Azure AI Search (formerly Azure Cognitive Search) is a cloud search service with vector search capabilities for RAG and semantic search applications. Use this node to store, retrieve, and query vector embeddings alongside their content and metadata.

On this page, you’ll find the node parameters for the Azure AI Search Vector Store node, and links to more resources.

-8<- “ snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Prerequisites

Before using this node, you need:

1. An [Azure subscription](#)
2. An [Azure AI Search service](#)
3. API key authentication configured (admin key for write operations, query key for read-only)

See [credentials documentation](#) for setup instructions.

Index configuration

The node automatically creates indexes if they don’t exist. When auto-creating, the node configures:

- Vector fields with appropriate dimensions based on your embeddings model
- HNSW algorithm for efficient similarity search with cosine metric
- Content and metadata fields for filtering and retrieval

You can also pre-create indexes in Azure Portal for custom configurations. Example schema:

```
{
  "name": "n8n-vectorstore",
  "fields": [
    {
      "name": "id",
      "type": "Edm.String",
      "key": true,
      "filterable": true
    },
    {
      "name": "content",
      "type": "Edm.String",
      "searchable": true
    }
  ]
}
```

```

        "name": "content_vector",
        "type": "Collection(Edm.Single)",
        "searchable": true,
        "vectorSearchDimensions": 1536,
        "vectorSearchProfileName": "n8n-vector-profile"
    },
    {
        "name": "metadata",
        "type": "Edm.String",
        "filterable": true
    }
],
"vectorSearch": {
    "profiles": [
        {
            "name": "n8n-vector-profile",
            "algorithm": "n8n-vector-algorithm"
        }
    ],
    "algorithms": [
        {
            "name": "n8n-vector-algorithm",
            "kind": "hnsw",
            "hnswParameters": {
                "metric": "cosine",
                "m": 4,
                "efConstruction": 400,
                "efSearch": 500
            }
        }
    ]
}

```

Node usage patterns

Use as a regular node to insert and retrieve documents

Use the node directly in workflows to insert or retrieve documents without an agent. See [this template](#) for an example pattern (uses Supabase, but the pattern is identical).

Connect directly to an AI agent as a tool

Connect to an [AI agent](#)'s tool connector to use the vector store as a searchable knowledge base:

AI agent (tools connector) → Azure AI Search Vector Store node

Use a retriever to fetch documents

Use with [Vector Store Retriever](#) and [Question and Answer Chain](#) for retrieval-augmented generation:

Question and Answer Chain (Retriever) → Vector Store Retriever (Vector Store) → Azure AI Search Vector Store

See [this example workflow](#).

Use the Vector Store Question Answer Tool

Use [Vector Store Question Answer Tool](#) to summarize and answer questions:

AI agent (tools) → Vector Store Question Answer Tool (Vector Store) → Azure AI Search Vector Store

See [this example](#).

Node parameters

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-mode-with-update.md”

Rerank Results

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-rerank-results.md”

Get Many parameters

- **Endpoint:** Your Azure AI Search endpoint (format: `https://your-service.search.windows.net`)
- **Index Name:** The index to query
- **Limit:** Maximum documents to return (default: 4)

Insert Documents parameters

- **Endpoint:** Your Azure AI Search endpoint
- **Index Name:** The index to use (created automatically if it doesn't exist)
- **Batch Size:** Number of documents uploaded per batch to Azure AI Search. Adjust based on document size and your service tier limits. This controls upload batching only—embedding generation batching is configured in embedding nodes.

Update Documents parameters

- **Endpoint:** Your Azure AI Search endpoint
- **Index Name:** The index to update

Retrieve Documents parameters (As Vector Store for Chain/Tool)

- **Endpoint:** Your Azure AI Search endpoint
- **Index Name:** The index to query

Retrieve Documents (As Tool for AI Agent) parameters

- **Name:** Tool name shown to the LLM

- **Description:** Explain to the LLM what this tool does. Be specific to help the LLM choose when to use this tool.
- **Endpoint:** Your Azure AI Search endpoint
- **Index Name:** The index to query
- **Limit:** Maximum results to retrieve (e.g., 10 for ten best matches)

Node options

Options

- **Filter:** [OData filter expression](#) to filter results by document fields or metadata. See filter examples below.
- **Query Mode:** Search strategy to use:
 - **Vector:** Similarity search using embeddings only
 - **Keyword:** Full-text search using BM25 ranking
 - **Hybrid** (default): Combines vector and keyword search with Reciprocal Rank Fusion (RRF)
 - **Semantic Hybrid:** Hybrid search with [semantic reranking](#) for improved relevance
- **Semantic Configuration:** Name of the semantic configuration to use for [semantic ranking](#). Defaults to semantic-search-config if not specified. Only required if you pre-created an index with a custom semantic configuration name.

OData filter examples

Azure AI Search uses [OData syntax](#) for filtering. Metadata fields are accessed using metadata/fieldName format.

Filter by document ID:

```
id eq '3da6491a-f930-4a4e-9471-c05dcd450ba0'
```

Filter by metadata field:

```
metadata/source eq 'user-guide'
```

Complex AND filter:

```
metadata/category eq 'technology' and metadata/author eq 'John'
```

Complex OR filter:

```
metadata/source eq 'user-guide' or metadata/rating ge 4
```

Numeric comparison:

```
metadata/rating ge 4 and metadata/rating lt 10
```

String matching with NOT:

```
metadata/category eq 'technology' and metadata/title ne 'Deprecated'
```

Supported OData operators: - Comparison: eq, ne, gt, ge, lt, le - Logical: and, or, not - String functions: startswith(), endswith(), contains() - Collection functions: any(), all()

Azure AI Search specific features

Hybrid search with RRF

Azure AI Search's hybrid search uses Reciprocal Rank Fusion to merge vector and keyword results, providing better accuracy than either method alone.

Semantic ranking

Semantic Hybrid mode applies machine learning models to rerank results based on semantic understanding of your query. This requires a semantic configuration in your index.

OData filters

Use OData syntax to filter by document fields or metadata before vector search executes. This improves performance and precision when you need results from specific sources or with certain attributes.

HNSW algorithm

Azure AI Search uses Hierarchical Navigable Small World (HNSW) graphs for approximate nearest neighbor search, providing fast retrieval at scale with configurable accuracy/speed tradeoffs.

Troubleshooting

Index issues

Index not found: Verify the index name is correct (case-sensitive) and exists in your Azure AI Search service. If using auto-creation, check that the index was created successfully.

Vector dimension mismatch: Ensure your embedding model dimensions match the index vector field dimensions. Check the index schema to confirm the `vectorSearchDimensions` setting.

Document insert failures: - Verify write permissions (admin API key required) - Check document fields match your index schema - Ensure required fields are provided in documents - Review batch size settings if experiencing timeouts with large document sets

Filter issues

Filter not working: - Verify OData syntax is correct - Ensure metadata fields use `metadata/` prefix: `metadata/source eq 'value'` - Check that filtered fields are marked as `filterable` in your index schema - Test with simple filters first (`id eq 'value'`) before complex expressions

Invalid OData syntax: - Use single quotes for string values: `metadata/source eq 'value'` - Use proper operators: `eq, ne, gt, ge, lt, le, and, or, not` - Refer to [OData filter documentation](#) for syntax details

Connection issues

Unable to connect: - Verify endpoint URL format: `https://your-service.search.windows.net` - Confirm your Azure AI Search service is running and accessible - Check network security groups, firewall rules, and private endpoint configurations - For Azure-hosted n8n, verify virtual network peering or service endpoint configuration if using private endpoints

Authentication issues

For authentication troubleshooting including API key errors, refer to the [credentials documentation troubleshooting section](#).

Templates and examples

Related resources

- [Azure AI Search Vector Search documentation](#)
- [LangChain Azure AI Search integration](#)
- [Azure AI Search REST API reference](#)
- [OData filter syntax for Azure AI Search](#)

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

-8<- “_snippets/self-hosting/starter-kits/self-hosted-ai-starter-kit.md”

Simple Vector Store node

Use the Simple Vector Store node to store and retrieve [embeddings](#) in n8n’s in-app memory.

On this page, you’ll find the node parameters for the Simple Vector Store node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Data safety limitations

Before using the Simple Vector Store node, it’s important to understand its limitations and how it works.

Vector store data isn’t persistent

This node stores data in memory only. All data is lost when n8n restarts and may also be purged in low-memory conditions.

All instance users can access vector store data

Memory keys for the Simple Vector Store node are global, not scoped to individual workflows.

This means that all users of the instance can access vector store data by adding a Simple Vector Store node and selecting the memory key, regardless of the access controls set for the original workflow. Take care not to expose sensitive information when ingesting data with the Simple Vector Store node.

Node usage patterns

You can use the Simple Vector Store node in the following patterns.

Use as a regular node to insert and retrieve documents

You can use the Simple Vector Store as a regular node to insert or get documents. This pattern places the Simple Vector Store in the regular connection flow without using an agent.

You can see an example of in step 2 of [this template](#).

Connect directly to an AI agent as a tool

You can connect the Simple Vector Store node directly to the [tool](#) connector of an [AI agent](#) to use a vector store as a resource when answering queries.

Here, the connection would be: AI agent (tools connector) -> Simple Vector Store node.

Use a retriever to fetch documents

You can use the [Vector Store Retriever](#) node with the Simple Vector Store node to fetch documents from the Simple Vector Store node. This is often used with the [Question and Answer Chain](#) node to fetch documents from the vector store that match the given chat input.

An [example of the connection flow](#) (the linked example uses Pinecone, but the pattern is the same) would be: Question and Answer Chain (Retriever connector) -> Vector Store Retriever (Vector Store connector) -> Simple Vector Store.

Use the Vector Store Question Answer Tool to answer questions

Another pattern uses the [Vector Store Question Answer Tool](#) to summarize results and answer questions from the Simple Vector Store node. Rather than connecting the Simple Vector Store directly as a tool, this pattern uses a tool specifically designed to summarizes data in the vector store.

The [connections flow](#) in this case would look like this: AI agent (tools connector) -> Vector Store Question Answer Tool (Vector Store connector) -> Simple Vector store.

Memory Management

The Simple Vector Store implements memory management to prevent excessive memory usage:

- Automatically cleans up old vector stores when memory pressure increases
- Removes inactive stores that haven't been accessed for a configurable amount of time

Configuration Options

You can control memory usage with these environment variables:

| Variable | Type | Default | Description |
|-----------------------------|--------|---------|---|
| N8N_VECTOR_STORE_MAX_MEMORY | Number | -1 | Maximum memory in MB allowed for all vector stores combined (-1 to disable limits). |
| N8N_VECTOR_STORE_TTL_HOURS | Number | -1 | Hours of inactivity after which a store gets removed (-1 to disable TTL). |

On n8n Cloud, these values are preset to 100MB (about 8,000 documents, depending on document size and metadata) and 7 days respectively. For self-hosted instances, both values default to -1(no memory limits or time-based cleanup).

Node parameters

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-mode.md”

Rerank Results

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-rerank-results.md”

Get Many parameters

- **Memory Key:** Select or create the key containing the vector memory you want to query.
- **Prompt:** Enter the search query.
- **Limit:** Enter how many results to retrieve from the vector store. For example, set this to 10 to get the ten best results.

Insert Documents parameters

- **Memory Key:** Select or create the key you want to store the vector memory as.
- **Clear Store:** Use this parameter to control whether to wipe the vector store for the given memory key for this workflow before inserting data (turned on).

Retrieve Documents (As Vector Store for Chain/Tool) parameters

- **Memory Key:** Select or create the key containing the vector memory you want to query.

Retrieve Documents (As Tool for AI Agent) parameters

- **Name:** The name of the vector store.
- **Description:** Explain to the LLM what this tool does. A good, specific description allows LLMs to produce expected results more often.
- **Memory Key:** Select or create the key containing the vector memory you want to query.
- **Limit:** Enter how many results to retrieve from the vector store. For example, set this to 10 to get the ten best results.

Templates and examples

Related resources

Refer to [LangChains's Memory Vector Store documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Milvus Vector Store node

Use the Milvus node to interact with your Milvus database as [vector store](#). You can insert documents into a vector database, get documents from a vector database, retrieve documents to provide them to a retriever connected to a [chain](#), or connect directly to an [agent](#) as a [tool](#).

On this page, you'll find the node parameters for the Milvus node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node usage patterns

You can use the Milvus Vector Store node in the following patterns.

Use as a regular node to insert and retrieve documents

You can use the Milvus Vector Store as a regular node to insert, or get documents. This pattern places the Milvus Vector Store in the regular connection flow without using an agent.

See this [example template](#) for how to build a system that stores documents in Milvus and retrieves them to support cited, chat-based answers.

Connect directly to an AI agent as a tool

You can connect the Milvus Vector Store node directly to the tool connector of an [AI agent](#) to use a vector store as a resource when answering queries.

Here, the connection would be: AI agent (tools connector) -> Milvus Vector Store node. See this [example template](#) where data is embedded and indexed in Milvus, and the AI Agent uses the vector store as a knowledge tool for question-answering.

Use a retriever to fetch documents

You can use the [Vector Store Retriever](#) node with the Milvus Vector Store node to fetch documents from the Milvus Vector Store node. This is often used with the [Question and Answer Chain](#) node to fetch documents from the vector store that match the given chat input.

A typical node connection flow looks like this: Question and Answer Chain (Retriever connector) -> Vector Store Retriever (Vector Store connector) -> Milvus Vector Store.

Check out this [workflow example](#) to see how to ingest external data into Milvus and build a chat-based semantic Q&A system.

Use the Vector Store Question Answer Tool to answer questions

Another pattern uses the [Vector Store Question Answer Tool](#) to summarize results and answer questions from the Milvus Vector Store node. Rather than connecting the Milvus Vector Store directly as a tool, this pattern uses a tool specifically designed to summarizes data in the vector store.

The connections flow would look like this: AI agent (tools connector) -> Vector Store Question Answer Tool (Vector Store connector) -> Milvus Vector store.

Node parameters

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-mode.md”

Rerank Results

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-rerank-results.md”

Get Many parameters

- **Milvus Collection:** Select or enter the Milvus Collection to use.
- **Prompt:** Enter your search query.

- **Limit:** Enter how many results to retrieve from the vector store. For example, set this to 10 to get the ten best results.

Insert Documents parameters

- **Milvus Collection:** Select or enter the Milvus Collection to use.
- **Clear Collection:** Specify whether to clear the collection before inserting new documents.

Retrieve Documents (As Vector Store for Chain/Tool) parameters

- **Milvus collection:** Select or enter the Milvus Collection to use.

Retrieve Documents (As Tool for AI Agent) parameters

- **Name:** The name of the vector store.
- **Description:** Explain to the LLM what this tool does. A good, specific description allows LLMs to produce expected results more often.
- **Milvus Collection:** Select or enter the Milvus Collection to use.
- **Limit:** Enter how many results to retrieve from the vector store. For example, set this to 10 to get the ten best results.

Node options

Metadata Filter

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/vector-store-metadata-filter.md"
```

Clear Collection

Available in **Insert Documents** mode. Deletes all data from the collection before inserting the new data.

Related resources

Refer to [LangChain's Milvus documentation](#) for more information about the service.

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

MongoDB Atlas Vector Store node

MongoDB Atlas Vector Search is a feature of MongoDB Atlas that enables users to store and query vector embeddings. Use this node to interact with Vector Search indexes in your MongoDB Atlas

collections. You can insert documents, retrieve documents, and use the vector store in chains or as a tool for agents.

On this page, you'll find the node parameters for the MongoDB Atlas Vector Store node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Prerequisites

Before using this node, create a [Vector Search index](#) in your MongoDB Atlas collection. Follow these steps to create one:

1. Log in to the [MongoDB Atlas dashboard](#).
2. Select your organization and project.
3. Find “Search & Vector Search” section.
4. Select your cluster and click “Go to search”.
5. Click “Create Search Index”.
6. Choose “Vector Search” mode and use the visual or JSON editors.
For example:

```
{  
  "fields": [  
    {  
      "type": "vector",  
      "path": "<field-name>",  
      "numDimensions": 1536, // any other value  
      "similarity": "<similarity-function>"  
    }  
  ]  
}
```

7. Adjust the “dimensions” value according to your embedding model (For example, 1536 for OpenAI’s text-embedding-small-3).
8. Name your index and create.

Make sure to note the following values which are required when configuring the node:

- Collection name
- Vector index name
- Field names for embeddings and metadata

Node usage patterns

You can use the MongoDB Atlas Vector Store node in the following patterns:

Use as a regular node to insert and retrieve documents

You can use the MongoDB Atlas Vector Store as a regular node to insert or get documents. This pattern places the MongoDB Atlas Vector Store in the regular connection flow without using an agent.

You can see an example of this in scenario 1 of [this template](#) (the template uses the Supabase Vector Store, but the pattern is the same).

Connect directly to an AI agent as a tool

You can connect the MongoDB Atlas Vector Store node directly to the tool connector of an [AI agent](#) to use the vector store as a resource when answering queries.

Here, the connection would be: AI agent (tools connector) -> MongoDB Atlas Vector Store node.

Use a retriever to fetch documents

You can use the [Vector Store Retriever](#) node with the MongoDB Atlas Vector Store node to fetch documents from the MongoDB Atlas Vector Store node. This is often used with the [Question and Answer Chain](#) node to fetch documents from the vector store that match the given chat input.

An [example of the connection flow](#) (the linked example uses Pinecone, but the pattern is the same) would be: Question and Answer Chain (Retriever connector) -> Vector Store Retriever (Vector Store connector) -> MongoDB Atlas Vector Store.

Use the Vector Store Question Answer Tool to answer questions

Another pattern uses the [Vector Store Question Answer Tool](#) to summarize results and answer questions from the MongoDB Atlas Vector Store node. Rather than connecting the MongoDB Atlas Vector Store directly as a tool, this pattern uses a tool specifically designed to summarize data in the vector store.

The [connections flow](#) (the linked example uses the In-Memory Vector Store, but the pattern is the same) in this case would look like this: AI agent (tools connector) -> Vector Store Question Answer Tool (Vector Store connector) -> In-Memory Vector store.

Node parameters

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-mode.md”

Rerank Results

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-rerank-results.md”

Get Many parameters

- **Mongo Collection:** Enter the name of the MongoDB collection to use.
- **Vector Index Name:** Enter the name of the Vector Search index in your MongoDB Atlas collection.
- **Embedding Field:** Enter the field name in your documents that contains the vector embeddings.
- **Metadata Field:** Enter the field name in your documents that contains the text metadata.

Insert Documents parameters

- **Mongo Collection:** Enter the name of the MongoDB collection to use.
- **Vector Index Name:** Enter the name of the Vector Search index in your MongoDB Atlas collection.
- **Embedding Field:** Enter the field name in your documents that contains the vector embeddings.
- **Metadata Field:** Enter the field name in your documents that contains the text metadata.

Retrieve Documents parameters (As Vector Store for Chain/Tool)

- **Mongo Collection:** Enter the name of the MongoDB collection to use.
- **Vector Index Name:** Enter the name of the Vector Search index in your MongoDB Atlas collection.
- **Embedding Field:** Enter the field name in your documents that contains the vector embeddings.
- **Metadata Field:** Enter the field name in your documents that contains the text metadata.

Retrieve Documents (As Tool for AI Agent) parameters

- **Name:** The name of the vector store.
- **Description:** Explain to the LLM what this tool does. A good, specific description allows LLMs to produce expected results more often.
- **Mongo Collection:** Enter the name of the MongoDB collection to use.
- **Vector Index Name:** Enter the name of the Vector Search index in your MongoDB Atlas collection.
- **Limit:** Enter how many results to retrieve from the vector store. For example, set this to 10 to get the ten best results.

Node options

Options

- **Metadata Filter:** Filters results based on metadata.

Templates and examples

Related resources

Refer to:

- [LangChain's MongoDB Atlas Vector Search documentation](#) for more information about the service.
- [MongoDB Atlas Vector Search documentation](#) for more information about MongoDB Atlas Vector Search.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

-8<- “_snippets/self-hosting/starter-kits/self-hosted-ai-starter-kit.md”

PGVector Vector Store node

PGVector is an extension of Postgresql. Use this node to interact with the PGVector tables in your Postgresql database. You can insert documents into a vector table, get documents from a vector table, retrieve documents to provide them to a retriever connected to a [chain](#), or connect directly to an [agent](#) as a [tool](#).

On this page, you'll find the node parameters for the PGVector node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node usage patterns

You can use the PGVector Vector Store node in the following patterns.

Use as a regular node to insert and retrieve documents

You can use the PGVector Vector Store as a regular node to insert or get documents. This pattern places the PGVector Vector Store in the regular connection flow without using an agent.

You can see an example of this in scenario 1 of [this template](#) (the template uses the Supabase Vector Store, but the pattern is the same).

Connect directly to an AI agent as a tool

You can connect the PGVector Vector Store node directly to the tool connector of an [AI agent](#) to use a vector store as a resource when answering queries.

Here, the connection would be: AI agent (tools connector) -> PGVector Vector Store node.

Use a retriever to fetch documents

You can use the [Vector Store Retriever](#) node with the PGVector Vector Store node to fetch documents from the PGVector Vector Store node. This is often used with the [Question and Answer Chain](#) node to fetch documents from the vector store that match the given chat input.

An [example of the connection flow](#) (the linked example uses Pinecone, but the pattern is the same) would be: Question and Answer Chain (Retriever connector) -> Vector Store Retriever (Vector Store connector) -> PGVector Vector Store.

Use the Vector Store Question Answer Tool to answer questions

Another pattern uses the [Vector Store Question Answer Tool](#) to summarize results and answer questions from the PGVector Vector Store node. Rather than connecting the PGVector Vector Store directly as a tool, this pattern uses a tool specifically designed to summarize data in the vector store.

The [connections flow](#) (the linked example uses the Simple Vector Store, but the pattern is the same) in this case would look like this: AI agent (tools connector) -> Vector Store Question Answer Tool (Vector Store connector) -> Simple Vector store.

Node parameters

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-mode.md”

Rerank Results

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-rerank-results.md”

Get Many parameters

- **Table name:** Enter the name of the table you want to query.
- **Prompt:** Enter your search query.
- **Limit:** Enter a number to set how many results to retrieve from the vector store. For example, set this to 10 to get the ten best results.

Insert Documents parameters

- **Table name:** Enter the name of the table you want to query.

Retrieve Documents parameters (As Vector Store for Chain/Tool)

- **Table name:** Enter the name of the table you want to query.

Retrieve Documents (As Tool for AI Agent) parameters

- **Name:** The name of the vector store.

- **Description:** Explain to the LLM what this tool does. A good, specific description allows LLMs to produce expected results more often.
- **Table Name:** Enter the PGVector table to use.
- **Limit:** Enter how many results to retrieve from the vector store. For example, set this to 10 to get the ten best results.

Node options

Collection

A way to separate datasets in PGVector. This creates a separate table and column to keep track of which collection a vector belongs to.

- **Use Collection:** Select whether to use a collection (turned on) or not (turned off).
- **Collection Name:** Enter the name of the collection you want to use.
- **Collection Table Name:** Enter the name of the table to store collection information in.

Column Names

The following options specify the names of the columns to store the vectors and corresponding information in:

- **ID Column Name**
- **Vector Column Name**
- **Content Column Name**
- **Metadata Column Name**

Metadata Filter

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/vector-store-metadata-filter.md”

Templates and examples

Related resources

Refer to [LangChain’s PGVector documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

-8<- “_snippets/self-hosting/starter-kits/self-hosted-ai-starter-kit.md”

Pinecone Vector Store node

Use the Pinecone node to interact with your Pinecone database as vector store. You can insert documents into a vector database, get documents from a vector database, retrieve documents to provide them to a retriever connected to a chain, or connect directly to an agent as a tool. You can also update an item in a vector database by its ID.

On this page, you'll find the node parameters for the Pinecone node, and links to more resources.

```
-8<- "_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node usage patterns

You can use the Pinecone Vector Store node in the following patterns.

Use as a regular node to insert, update, and retrieve documents

You can use the Pinecone Vector Store as a regular node to insert, update, or get documents. This pattern places the Pinecone Vector Store in the regular connection flow without using an agent.

You can see an example of this in scenario 1 of [this template](#).

Connect directly to an AI agent as a tool

You can connect the Pinecone Vector Store node directly to the tool connector of an AI agent to use a vector store as a resource when answering queries.

Here, the connection would be: AI agent (tools connector) -> Pinecone Vector Store node.

Use a retriever to fetch documents

You can use the Vector Store Retriever node with the Pinecone Vector Store node to fetch documents from the Pinecone Vector Store node. This is often used with the Question and Answer Chain node to fetch documents from the vector store that match the given chat input.

An [example of the connection flow](#) would be: Question and Answer Chain (Retriever connector) -> Vector Store Retriever (Vector Store connector) -> Pinecone Vector Store.

Use the Vector Store Question Answer Tool to answer questions

Another pattern uses the Vector Store Question Answer Tool to summarize results and answer questions from the Pinecone Vector Store node. Rather than connecting the Pinecone Vector Store directly as a tool, this pattern uses a tool specifically designed to summarizes data in the vector store.

The connections flow in this case would look like this: AI agent (tools connector) -> Vector Store Question Answer Tool (Vector Store connector) -> Pinecone Vector store.

Node parameters

Operation Mode

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-mode-with-update.md”

Rerank Results

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-rerank-results.md”

Get Many parameters

- **Pinecone Index:** Select or enter the Pinecone Index to use.
- **Prompt:** Enter your search query.
- **Limit:** Enter how many results to retrieve from the vector store. For example, set this to 10 to get the ten best results.

Insert Documents parameters

- **Pinecone Index:** Select or enter the Pinecone Index to use.

Retrieve Documents (As Vector Store for Chain/Tool) parameters

- **Pinecone Index:** Select or enter the Pinecone Index to use.

Retrieve Documents (As Tool for AI Agent) parameters

- **Name:** The name of the vector store.
- **Description:** Explain to the LLM what this tool does. A good, specific description allows LLMs to produce expected results more often.
- **Pinecone Index:** Select or enter the Pinecone Index to use.
- **Limit:** Enter how many results to retrieve from the vector store. For example, set this to 10 to get the ten best results.

Parameters for Update Documents

- ID

Node options

Pinecone Namespace

Another segregation option for how to store your data within the index.

Metadata Filter

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/vector-store-metadata-filter.md"
```

Clear Namespace

Available in **Insert Documents** mode. Deletes all data from the namespace before inserting the new data.

Templates and examples

Related resources

Refer to [LangChain's Pinecone documentation](#) for more information about the service.

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

Find your Pinecone index and namespace

Your Pinecone index and namespace are available in your Pinecone account.

[Image: Screenshot of a Pinecone account, with the Pinecone index labelled]

Qdrant Vector Store node

Use the Qdrant node to interact with your Qdrant collection as a vector store. You can insert documents into a vector database, get documents from a vector database, retrieve documents to provide them to a retriever connected to a chain or connect it directly to an agent to use as a tool.

On this page, you'll find the node parameters for the Qdrant node, and links to more resources.

```
-8<- "snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node usage patterns

You can use the Qdrant Vector Store node in the following patterns.

Use as a regular node to insert and retrieve documents

You can use the Qdrant Vector Store as a regular node to insert or get documents. This pattern places the Qdrant Vector Store in the regular connection flow without using an agent.

You can see an example of this in the first part of [this template](#).

Connect directly to an AI agent as a tool

You can connect the Qdrant Vector Store node directly to the tool connector of an [AI agent](#) to use a vector store as a resource when answering queries.

Here, the connection would be: AI agent (tools connector) -> Qdrant Vector Store node.

Use a retriever to fetch documents

You can use the [Vector Store Retriever](#) node with the Qdrant Vector Store node to fetch documents from the Qdrant Vector Store node. This is often used with the [Question and Answer Chain](#) node to fetch documents from the vector store that match the given chat input.

An [example of the connection flow](#) would be: Question and Answer Chain (Retriever connector) -> Vector Store Retriever (Vector Store connector) -> Qdrant Vector Store.

Use the Vector Store Question Answer Tool to answer questions

Another pattern uses the [Vector Store Question Answer Tool](#) to summarize results and answer questions from the Qdrant Vector Store node. Rather than connecting the Qdrant Vector Store directly as a tool, this pattern uses a tool specifically designed to summarizes data in the vector store.

The [connections flow](#) in this case would look like this: AI agent (tools connector) -> Vector Store Question Answer Tool (Vector Store connector) -> Qdrant Vector store.

Node parameters

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-mode.md”

Rerank Results

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-rerank-results.md”

Get Many parameters

- **Qdrant collection name:** Enter the name of the Qdrant collection to use.
- **Prompt:** Enter the search query.
- **Limit:** Enter how many results to retrieve from the vector store. For example, set this to 10 to get the ten best results.

This Operation Mode includes one **Node option**, the [Metadata Filter](#).

Insert Documents parameters

- **Qdrant collection name:** Enter the name of the Qdrant collection to use.

This Operation Mode includes one **Node option**:

- **Collection Config:** Enter JSON options for creating a Qdrant collection creation configuration. Refer to the Qdrant [Collections](#) documentation for more information.

Retrieve Documents (As Vector Store for Chain/Tool) parameters

- **Qdrant Collection:** Enter the name of the Qdrant collection to use.

This Operation Mode includes one **Node option**, the [Metadata Filter](#).

Retrieve Documents (As Tool for AI Agent) parameters

- **Name:** The name of the vector store.
- **Description:** Explain to the LLM what this tool does. A good, specific description allows LLMs to produce expected results more often.
- **Qdrant Collection:** Enter the name of the Qdrant collection to use.
- **Limit:** Enter how many results to retrieve from the vector store. For example, set this to 10 to get the ten best results.

Node options

Metadata Filter

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/vector-store-metadata-filter.md"
```

Templates and examples

Related resources

Refer to [LangChain's Qdrant documentation](#) for more information about the service.

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

```
-8<- "_snippets/self-hosting/starter-kits/self-hosted-ai-starter-kit.md"
```

Redis Vector Store node

Use the Redis Vector Store node to interact with your Redis database as a [vector store](#). You can insert documents into the vector database, get documents from the vector database, retrieve documents using a retriever connected to a [chain](#), or connect it directly to an [agent](#) to use as a [tool](#).

On this page, you'll find the node parameters for the Redis Vector Store node, and links to more resources.

-8<- “ snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Prerequisites

Before using this node, you need a Redis database with the [Redis Query Engine](#) enabled. Use one of the following:

- Redis Open Source (v8.0 and later) - includes the Redis Query Engine by default
- [Redis Cloud](#) - fully managed service
- [Redis Software](#) - self-managed deployment

Node usage patterns

You can use the Redis Vector Store node in the following patterns:

Use as a regular node to insert and retrieve documents

You can use the Redis Vector Store as a regular node to insert or get documents. This pattern places the Redis Vector Store in the regular connection flow without using an agent.

You can see an example of this in scenario 1 of [this template](#) (the template uses the Supabase Vector Store, but the pattern is the same).

Connect directly to an AI agent as a tool

You can connect the Redis Vector Store node directly to the [tool](#) connector of an [AI agent](#) to use a vector store as a resource when answering queries.

Here, the connection would be: AI agent (tools connector) -> Redis Vector Store node.

Use a retriever to fetch documents

You can use the [Vector Store Retriever](#) node with the Redis Vector Store node to fetch documents from the Redis Vector Store node. This is often used with the [Question and Answer Chain](#) node to fetch documents from the vector store that match the given chat input.

An [example of the connection flow](#) (the linked example uses Pinecone, but the pattern is the same) would be: Question and Answer Chain (Retriever connector) -> Vector Store Retriever (Vector Store

connector) -> Redis Vector Store.

Use the Vector Store Question Answer Tool to answer questions

Another pattern uses the [Vector Store Question Answer Tool](#) to summarize results and answer questions from the Redis Vector Store node. Rather than connecting the Redis Vector Store directly as a tool, this pattern uses a tool specifically designed to summarizes data in the vector store.

The [connections flow](#) (the linked example uses Qdrant, but the pattern is the same) in this case would look like this: AI agent (tools connector) -> Vector Store Question Answer Tool (Vector Store connector) -> Redis Vector store.

Node parameters

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-mode.md”

Rerank Results

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-rerank-results.md”

Get Many parameters

- **Redis Index:** Enter the name of the Redis vector search index to use. Optionally choose an existing one from the list.
- **Prompt:** Enter the search query.
- **Limit:** Enter how many results to retrieve from the vector store. For example, set this to 10 to get the ten best results.

This Operation Mode includes one **Node option**, the [Metadata Filter](#).

Insert Documents parameters

- **Redis Index:** Enter the name of the Redis vector search index to use. Optionally choose an existing one from the list.

Retrieve Documents (As Vector Store for Chain/Tool) parameters

- **Redis Index:** Enter the name of the Redis vector search index to use.

This Operation Mode includes one **Node option**, the [Metadata Filter](#). Optionally choose an existing one from the list.

Retrieve Documents (As Tool for AI Agent) parameters

- **Name:** The name of the vector store.
- **Description:** Explain to the LLM what this tool does. A good,

specific description allows LLMs to produce expected results more often.

- **Redis Index:** Enter the name of the Redis vector search index to use. Optionally choose an existing one from the list.
- **Limit:** Enter how many results to retrieve from the vector store. For example, set this to 10 to get the ten best results.

Include Metadata

Whether to include document metadata.

You can use this with the [Get Many](#) and [Retrieve Documents \(As Tool for AI Agent\)](#) modes.

Node options

Metadata Filter

Metadata filters are available for the [Get Many](#), [Retrieve Documents \(As Vector Store for Chain/Tool\)](#), and [Retrieve Documents \(As Tool for AI Agent\)](#) operation modes. This is an OR query. If you specify more than one metadata filter field, at least one of them must match. When inserting data, the metadata is set using the document loader. Refer to [Default Data Loader](#) for more information on loading documents.

Redis Configuration Options

Available for all operation modes:

- **Metadata Key:** Enter the key for the metadata field in the Redis hash (default: `metadata`).
- **Key Prefix:** Enter the key prefix for storing documents (default: `doc:`).
- **Content Key:** Enter the key for the content field in the Redis hash (default: `content`).
- **Embedding Key:** Enter the key for the embedding field in the Redis hash (default: `embedding`).

Insert Options

Available for the [Insert Documents](#) operation mode:

- **Overwrite Documents:** Select whether to overwrite existing documents (turned on) or not (turned off). Also deletes the index.
- **Time-to-Live:** Enter the time-to-live for documents in seconds. Does not expire the index.

Templates and examples

Related resources

Refer to:

- [Redis Vector Search documentation](#) for more information about

Redis vector capabilities.

- [RediSearch documentation](#) for more information about RediSearch.
- [LangChain's Redis Vector Store documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

-8<- “_snippets/self-hosting/starter-kits/self-hosted-ai-starter-kit.md”

Supabase Vector Store node

Use the Supabase Vector Store to interact with your Supabase database as vector store. You can insert documents into a vector database, get many documents from a vector database, and retrieve documents to provide them to a retriever connected to a chain.

Use the Supabase Vector Store to interact with your Supabase database as [vector store](#). You can insert documents into a vector database, get documents from a vector database, retrieve documents to provide them to a retriever connected to a [chain](#), or connect it directly to an [agent](#) to use as a [tool](#). You can also update an item in a vector store by its ID.

On this page, you'll find the node parameters for the Supabase node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Supabase provides a [quickstart for setting up your vector store](#). If you use settings other than the defaults in the quickstart, this may affect parameter settings in n8n. Make sure you understand what you're doing.

Node usage patterns

You can use the Supabase Vector Store node in the following patterns.

Use as a regular node to insert, update, and retrieve documents

You can use the Supabase Vector Store as a regular node to insert, update, or get documents. This pattern places the Supabase Vector Store in the regular connection flow without using an agent.

You can see an example of this in scenario 1 of [this template](#).

Connect directly to an AI agent as a tool

You can connect the Supabase Vector Store node directly to the tool connector of an [AI agent](#) to use a vector store as a resource when answering queries.

Here, the connection would be: AI agent (tools connector) -> Supabase Vector Store node.

Use a retriever to fetch documents

You can use the [Vector Store Retriever](#) node with the Supabase Vector Store node to fetch documents from the Supabase Vector Store node. This is often used with the [Question and Answer Chain](#) node to fetch documents from the vector store that match the given chat input.

An [example of the connection flow](#) (the example uses Pinecone, but the pattern in the same) would be: Question and Answer Chain (Retriever connector) -> Vector Store Retriever (Vector Store connector) -> Supabase Vector Store.

Use the Vector Store Question Answer Tool to answer questions

Another pattern uses the [Vector Store Question Answer Tool](#) to summarize results and answer questions from the Supabase Vector Store node. Rather than connecting the Supabase Vector Store directly as a tool, this pattern uses a tool specifically designed to summarizes data in the vector store.

The [connections flow](#) in this case would look like this: AI agent (tools connector) -> Vector Store Question Answer Tool (Vector Store connector) -> Supabase Vector store.

Node parameters

Operation Mode

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-mode-with-update.md”

Rerank Results

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-rerank-results.md”

Get Many parameters

- **Table Name:** Enter the Supabase table to use.
- **Prompt:** Enter the search query.
- **Limit:** Enter how many results to retrieve from the vector store.
For example, set this to 10 to get the ten best results.

Insert Documents parameters

- **Table Name:** Enter the Supabase table to use.

Retrieve Documents (As Vector Store for Chain/Tool) parameters

- **Table Name:** Enter the Supabase table to use.

Retrieve Documents (As Tool for AI Agent) parameters

- **Name:** The name of the vector store.
- **Description:** Explain to the LLM what this tool does. A good, specific description allows LLMs to produce expected results more often.
- **Table Name:** Enter the Supabase table to use.
- **Limit:** Enter how many results to retrieve from the vector store. For example, set this to 10 to get the ten best results.

Update Documents

- **Table Name:** Enter the Supabase table to use.
- **ID:** The ID of an embedding entry.

Parameters for **Update Documents**

- ID

Node options

Query Name

The name of the matching function you set up in Supabase. If you follow the [Supabase quickstart](#), this will be `match_documents`.

Metadata Filter

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/vector-store-metadata-filter.md"
```

Templates and examples

Related resources

Refer to [LangChain's Supabase documentation](#) for more information about the service.

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

Weaviate Vector Store node

Use the Weaviate node to interact with your Weaviate collection as a [vector store](#). You can insert documents into or retrieve documents from a vector database. You can also retrieve documents to provide them to a retriever connected to a [chain](#) or connect this node directly to an [agent](#) to use as a [tool](#). On this page, you'll find the node parameters for the Weaviate node, and links to more resources.

```
-8<- "snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node usage patterns

You can use the Weaviate Vector Store node in the following patterns.

Use as a regular node to insert and retrieve documents

You can use the Weaviate Vector Store as a regular node to insert or get documents. This pattern places the Weaviate Vector Store in the regular connection flow without using an agent.

Connect directly to an AI agent as a tool

You can connect the Weaviate Vector Store node directly to the tool connector of an [AI agent](#) to use a vector store as a resource when answering queries.

Here, the connection would be: AI agent (tools connector) -> Weaviate Vector Store node.

Use a retriever to fetch documents

You can use the [Vector Store Retriever](#) node with the Weaviate Vector Store node to fetch documents from the Weaviate Vector Store node. This is often used with the [Question and Answer Chain](#) node to fetch documents from the vector store that match the given chat input.

Use the Vector Store Question Answer Tool to answer questions

Another pattern uses the [Vector Store Question Answer Tool](#) to summarize results and answer questions from the Weaviate Vector Store node. Rather than connecting the Weaviate Vector Store directly as a tool, this pattern uses a tool specifically designed to summarizes data in the vector store.

Node parameters

```
-8<- "snippets/integrations/builtin/cluster-nodes/vector-store-mode.md"
```

Get Many parameters

- **Weaviate Collection:** Enter the name of the Weaviate collection to use.
- **Prompt:** Enter the search query.
- **Limit:** Enter how many results to retrieve from the vector store. For example, set this to 10 to get the ten best results.

Insert Documents parameters

- **Weaviate Collection:** Enter the name of the Weaviate collection to use.
- **Embedding Batch Size:** The number of documents to embed in a single batch. The default is 200 documents.

Retrieve Documents (As Vector Store for Chain/Tool) parameters

- **Weaviate Collection:** Enter the name of the Weaviate collection to use.

Retrieve Documents (As Tool for AI Agent) parameters

- **Weaviate Collection:** The name of the vector store.
- **Description:** Explain to the LLM what this tool does. A good, specific description allows LLMs to produce expected results more often.
- **Weaviate Collection:** Enter the name of the Weaviate collection to use.
- **Limit:** Enter how many results to retrieve from the vector store. For example, set this to 10 to get the ten best results.

Include Metadata

Whether to include document metadata.

You can use this with the [Get Many](#) and [Retrieve Documents \(As Tool for AI Agent\)](#) modes.

Rerank Results

```
-8<- "_snippets/integrations/builtin/cluster-nodes/vector-store-rerank-results.md"
```

Node options

Search Filters

Available for the [Get Many](#), [Retrieve Documents \(As Vector Store for Chain/Tool\)](#), and [Retrieve Documents \(As Tool for AI Agent\)](#) operation modes.

When searching for data, use this to match metadata associated with documents. You can learn more about the operators and query structure in [Weaviate's conditional filters documentation](#).

You can use both AND and OR with different operators. Operators are case insensitive:

```
{
  "OR": [
    {
      "path": ["source"],
      "operator": "Equal",
      "valueString": "source1"
    }
  ]
}
```

```

    },
{
  "path": ["source"],
  "operator": "Equal",
  "valueString": "source1"
}
]
}

```

Supported operators:

| Operator | Required Field(s) | Description |
|------------------|--|--|
| 'equal' | valueString or valueNumber | Checks if the property is equal to the given string or number. |
| 'like' | valueString | Checks if the string property matches a pattern (for example, sub-string match). |
| 'containsAny' | valueTextArray (string[]) | Checks if the property contains any of the given values. |
| 'containsAll' | valueTextArray (string[]) | Checks if the property contains all of the given values. |
| 'greaterThan' | valueNumber | Checks if the property value is greater than the given number. |
| 'lessThan' | valueNumber | Checks if the property value is less than the given number. |
| 'isNull' | valueBoolean (true/false) | Checks if the property is null or not. (<u>must enable before ingestion</u>) |
| 'withinGeoRange' | valueGeoCoordinates (object with geolocation data) | Filters by proximity to geographic coordinates. |

When inserting data, the document loader sets the metadata. Refer to [Default Data Loader](#) for more information on loading documents.

Metadata Keys

You can define which metadata keys you want Weaviate to return on your queries. This can reduce network load, as you will only get properties you have defined. Returns all properties from the server by default.

Available for the [Get Many](#), [Retrieve Documents \(As Vector Store for Chain/Tool\)](#), and [Retrieve Documents \(As Tool for AI Agent\)](#) operation modes.

Tenant Name

The specific tenant to store or retrieve documents for.

Text Key

The key in the document that contains the embedded text.

Skip Init Checks

Whether to skip initialization checks when instantiating the client.

Init Timeout

Number of seconds to wait before timing out during initial checks.

Insert Timeout

Number of seconds to wait before timing out during inserts.

Query Timeout

Number of seconds to wait before timing out during queries.

GRPC Proxy

A proxy to use for gRPC requests.

Clear Data

Available for the Insert Documents operation mode.

Whether to clear the collection or tenant before inserting new data.

Templates and examples

Related resources

Refer to LangChain's Weaviate documentation for more information about the service.

Refer to Weaviate Installation for a self hosted Weaviate Cluster.

-8<- “ snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Zep Vector Store node

Use the Zep Vector Store to interact with Zep vector databases. You can insert documents into a vector database, get documents from a vector database, retrieve documents to provide them to a retriever connected to a chain, or connect it directly to an agent to use as a tool.

On this page, you'll find the node parameters for the Zep Vector Store node, and links to more resources.

```
-8<- "_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node usage patterns

You can use the Zep Vector Store node in the following patterns.

Use as a regular node to insert, update, and retrieve documents

You can use the Zep Vector Store as a regular node to insert or get documents. This pattern places the Zep Vector Store in the regular connection flow without using an agent.

You can see an example of this in scenario 1 of [this template](#) (the example uses Supabase, but the pattern is the same).

Connect directly to an AI agent as a tool

You can connect the Zep Vector Store node directly to the tool connector of an [AI agent](#) to use a vector store as a resource when answering queries.

Here, the connection would be: AI agent (tools connector) -> Zep Vector Store node.

Use a retriever to fetch documents

You can use the [Vector Store Retriever](#) node with the Zep Vector Store node to fetch documents from the Zep Vector Store node. This is often used with the [Question and Answer Chain](#) node to fetch documents from the vector store that match the given chat input.

An [example of the connection flow](#) (the example uses Pinecone, but the pattern is the same) would be: Question and Answer Chain (Retriever connector) -> Vector Store Retriever (Vector Store connector) -> Zep Vector Store.

Use the Vector Store Question Answer Tool to answer questions

Another pattern uses the [Vector Store Question Answer Tool](#) to summarize results and answer questions from the Zep Vector Store node. Rather than connecting the Zep Vector Store directly as a tool, this pattern uses a tool specifically designed to summarizes data in the vector store.

The [connections flow](#) (this example uses Supabase, but the pattern is the same) in this case would look like this: AI agent (tools connector) -> Vector Store Question Answer Tool (Vector Store connector) -> Zep Vector store.

Node parameters

```
-8<- "_snippets/integrations/builtin/cluster-nodes/vector-store-mode.md"
```

Rerank Results

-8<- “_snippets/integrations/builtin/cluster-nodes/vector-store-rerank-results.md”

Insert Documents parameters

- **Collection Name:** Enter the collection name to store the data in.

Get Many parameters

- **Collection Name:** Enter the collection name to retrieve the data from.
- **Prompt:** Enter the search query.
- **Limit:** Enter how many results to retrieve from the vector store. For example, set this to 10 to get the ten best results.

Retrieve Documents (As Vector Store for Chain/Tool) parameters

- **Collection Name:** Enter the collection name to retrieve the data from.

Retrieve Documents (As Tool for AI Agent) parameters

- **Name:** The name of the vector store.
- **Description:** Explain to the LLM what this tool does. A good, specific description allows LLMs to produce expected results more often.
- **Collection Name:** Enter the collection name to retrieve the data from.
- **Limit:** Enter how many results to retrieve from the vector store. For example, set this to 10 to get the ten best results.

Node options

Embedding Dimensions

Must be the same when embedding the data and when querying it.

This sets the size of the array of floats used to represent the semantic meaning of a text document.

Is Auto Embedded

Available in the **Insert Documents** Operation Mode, enabled by default.

Disable this to configure your embeddings in Zep instead of in n8n.

Metadata Filter

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/vector-store-metadata-filter.md"
```

Templates and examples

Related resources

Refer to [LangChain's Zep documentation](#) for more information about the service.

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

Sub nodes

Sub nodes attach to root nodes within a group of cluster nodes. They configure the overall functionality of the cluster.

```
-8<- "snippets/integrations/builtin/cluster-nodes/cluster-nodes-summary.md"
```

Default Data Loader node

Use the Default Data Loader node to load binary data files or JSON data for [vector stores](#) or summarization.

On this page, you'll find a list of parameters the Default Data Loader node supports, and links to more resources.

```
-8<- "snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node parameters

- **Text Splitting:** Choose from:
 - **Simple:** Uses the [Recursive Character Text Splitter](#) with a chunk size of 1000 and an overlap of 200.
 - **Custom:** Allows you to connect a text splitter of your choice.
- **Type of Data:** Select **Binary** or **JSON**.
- **Mode:** Choose from:
 - **Load All Input Data:** Use all the node's input data.
 - **Load Specific Data:** Use [expressions](#) to define the data you want to load. You can add text as well as expressions. This means you can create a custom document from a mix of text and expressions.
- **Data Format:** Displays when you set **Type of Data** to **Binary**. Select the file MIME type for your binary data. Set to **Automatically Detect by MIME Type** if you want n8n to set the data format for you. If you set a specific data format and the incoming file MIME type doesn't match it, the node errors. If you use **Automatically Detect by MIME Type**, the node falls back to

text format if it can't match the file MIME type to a supported data format.

Node options

- **Metadata:** Set the metadata that should accompany the document in the vector store. This is what you match to using the **Metadata Filter** option when retrieving data using the vector store nodes.

Templates and examples

Related resources

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-doc-loaders-link.md”

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

GitHub Document Loader node

Use the GitHub Document Loader node to load data from a GitHub repository for vector stores or summarization.

On this page, you'll find the node parameters for the GitHub Document Loader node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Text Splitting:** Choose from:
 - **Simple:** Uses the Recursive Character Text Splitter with a chunk size of 1000 and an overlap of 200.
 - **Custom:** Allows you to connect a text splitter of your choice.
- **Repository Link:** Enter the URL of your GitHub repository.
- **Branch:** Enter the branch name to use.

Node options

- **Recursive:** Select whether to include sub-folders and files (turned on) or not (turned off).
- **Ignore Paths:** Enter directories to ignore.

Templates and examples

Related resources

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-doc-loaders-link.md"
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

Embeddings AWS Bedrock node

Use the Embeddings AWS Bedrock node to generate [embeddings](#) for a given text.

On this page, you'll find the node parameters for the Embeddings AWS Bedrock node, and links to more resources.

```
-8<- "_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node parameters

- **Model:** Select the model to use to generate the embedding.

Learn more about available models in the [Amazon Bedrock documentation](#).

Templates and examples

Related resources

Refer to [LangChains's AWS Bedrock embeddings documentation](#) and the [AWS Bedrock documentation](#) for more information about AWS Bedrock.

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

Embeddings Azure OpenAI node

Use the Embeddings Azure OpenAI node to generate [embeddings](#) for a given text.

On this page, you'll find the node parameters for the Embeddings Azure OpenAI node, and links to more resources.

```
-8<- "_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node options

- **Model (Deployment) Name:** Select the model (deployment) to use for generating embeddings.

- **Batch Size:** Enter the maximum number of documents to send in each request.
- **Strip New Lines:** Select whether to remove new line characters from input text (turned on) or not (turned off). n8n enables this by default.
- **Timeout:** Enter the maximum amount of time a request can take in seconds. Set to -1 for no timeout.

Templates and examples

Related resources

Refer to [LangChains's OpenAI embeddings documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Embeddings Cohere node

Use the Embeddings Cohere node to generate [embeddings](#) for a given text.

On this page, you'll find the node parameters for the Embeddings Cohere node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Model:** Select the model to use to generate the embedding.
Choose from:
 - **Embed-English-v2.0(4096 Dimensions)**
 - **Embed-English-Light-v2.0(1024 Dimensions)**
 - **Embed-Multilingual-v2.0(768 Dimensions)**

Learn more about available models in [Cohere's models documentation](#).

Templates and examples

Related resources

Refer to [Langchain's Cohere embeddings documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Embeddings Google Gemini node

Use the Embeddings Google Gemini node to generate [embeddings](#) for a given text.

On this page, you'll find the node parameters for the Embeddings Google Gemini node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Model:** Select the model to use to generate the embedding.

Learn more about available models in [Google Gemini's models documentation](#).

Templates and examples

Related resources

Refer to [Langchain's Google Generative AI embeddings documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Embeddings Google PaLM node

Use the Embeddings Google PaLM node to generate [embeddings](#) for a given text.

On this page, you'll find the node parameters for the Embeddings Google PaLM node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Model:** Select the model to use to generate the embedding.

n8n dynamically loads models from the Google PaLM API and you'll only see the models available to your account.

Templates and examples

Related resources

Refer to [Langchain's Google PaLM embeddings documentation](#) for more information about the service.

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

Embeddings Google Vertex node

Use the Embeddings Google Vertex node to generate [embeddings](#) for a given text.

On this page, you'll find the node parameters for the Embeddings Google Vertex node, and links to more resources.

```
-8<- "snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node parameters

- **Model:** Select the model to use to generate the embedding.

Learn more about available embedding models in [Google VertexAI embeddings API documentation](#).

Templates and examples

Related resources

Refer to [LangChain's Google Generative AI embeddings documentation](#) for more information about the service.

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

Embeddings HuggingFace Inference node

Use the Embeddings HuggingFace Inference node to generate [embeddings](#) for a given text.

On this page, you'll find the node parameters for the Embeddings HuggingFace Inference, and links to more resources.

```
-8<- "snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node parameters

- **Model:** Select the model to use to generate the embedding.

Refer to the [Hugging Face models documentation](#) for available models.

Node options

- **Custom Inference Endpoint:** Enter the URL of your deployed model, hosted by HuggingFace. If you set this, n8n ignores the **Model Name**.

Refer to [HuggingFace's guide to inference](#) for more information.

Templates and examples

Related resources

Refer to [Langchain's HuggingFace Inference embeddings documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Embeddings Mistral Cloud node

Use the Embeddings Mistral Cloud node to generate [embeddings](#) for a given text.

On this page, you'll find the node parameters for the Embeddings Mistral Cloud node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Model:** Select the model to use to generate the embedding.

Learn more about available models in [Mistral's models documentation](#).

Node options

- **Batch Size:** Enter the maximum number of documents to send in each request.
- **Strip New Lines:** Select whether to remove new line characters from input text (turned on) or not (turned off). n8n enables this by default.

Templates and examples

Related resources

Refer to [Langchain's Mistral embeddings documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Embeddings Ollama node

Use the Embeddings Ollama node to generate [embeddings](#) for a given text.

On this page, you'll find the node parameters for the Embeddings Ollama node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Model:** Select the model to use to generate the embedding.
Choose from:
 - [all-minilm](#) (384 Dimensions)
 - [nomic-embed-text](#) (768 Dimensions)

Learn more about available models in [Ollama's models documentation](#).

Templates and examples

Related resources

Refer to [Langchain's Ollama embeddings documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Embeddings OpenAI node

Use the Embeddings OpenAI node to generate [embeddings](#) for a given text.

On this page, you'll find the node parameters for the Embeddings OpenAI node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node options

- **Model:** Select the model to use for generating embeddings.
- **Base URL:** Enter the URL to send the request to. Use this if you are using a self-hosted OpenAI-like model.
- **Batch Size:** Enter the maximum number of documents to send in each request.
- **Strip New Lines:** Select whether to remove new line characters from input text (turned on) or not (turned off). n8n enables this by default.
- **Timeout:** Enter the maximum amount of time a request can take in seconds. Set to -1 for no timeout.

Templates and examples

Related resources

Refer to [LangChains's OpenAI embeddings documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Anthropic Chat Model node

Use the Anthropic Chat Model node to use Anthropic’s Claude family of chat models with [conversational agents](#).

On this page, you’ll find the node parameters for the Anthropic Chat Model node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Model:** Select the model that generates the completion. Choose from:
 - **Claude**
 - **Claude Instant**

Learn more in the [Anthropic model documentation](#).

Node options

- **Maximum Number of Tokens:** Enter the maximum number of tokens used, which sets the completion length.
- **Sampling Temperature:** Use this option to control the randomness of the sampling process. A higher temperature creates more diverse sampling, but increases the risk of hallucinations.
- **Top K:** Enter the number of token choices the model uses to generate the next token.

- **Top P:** Use this option to set the probability the completion should use. Use a lower value to ignore less probable options.

Templates and examples

Related resources

Refer to [LangChains's Anthropic documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

AWS Bedrock Chat Model node

The AWS Bedrock Chat Model node allows you use LLM models utilising AWS Bedrock platform.

On this page, you'll find the node parameters for the AWS Bedrock Chat Model node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Model:** Select the model that generates the completion.

Learn more about available models in the [Amazon Bedrock model documentation](#).

Node options

- **Maximum Number of Tokens:** Enter the maximum number of tokens used, which sets the completion length.
- **Sampling Temperature:** Use this option to control the randomness of the sampling process. A higher temperature creates more diverse sampling, but increases the risk of hallucinations.

Proxy limitations

This node doesn't support the `N0_PROXY` environment variable.

Templates and examples

Related resources

Refer to [LangChains's AWS Bedrock Chat Model documentation](#) for more information about the service.

-8<- “ snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Azure OpenAI Chat Model node

Use the Azure OpenAI Chat Model node to use OpenAI’s chat models with conversational [agents](#).

On this page, you’ll find the node parameters for the Azure OpenAI Chat Model node, and links to more resources.

-8<- “ snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Model:** Select the model to use to generate the completion.

Node options

- **Frequency Penalty:** Use this option to control the chances of the model repeating itself. Higher values reduce the chance of the model repeating itself.
- **Maximum Number of Tokens:** Enter the maximum number of tokens used, which sets the completion length.
- **Response Format:** Choose **Text** or **JSON**. **JSON** ensures the model returns valid JSON.
- **Presence Penalty:** Use this option to control the chances of the model talking about new topics. Higher values increase the chance of the model talking about new topics.
- **Sampling Temperature:** Use this option to control the randomness of the sampling process. A higher temperature creates more diverse sampling, but increases the risk of hallucinations.
- **Timeout:** Enter the maximum request time in milliseconds.
- **Max Retries:** Enter the maximum number of times to retry a request.
- **Top P:** Use this option to set the probability the completion should use. Use a lower value to ignore less probable options.

Proxy limitations

This node doesn’t support the [NO_PROXY](#) environment variable.

Templates and examples

Related resources

Refer to [LangChains’s Azure OpenAI documentation](#) for more information about the service.

-8<- “ snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Cohere Chat Model node

Use the Cohere Chat Model node to access Cohere’s large language models for conversational AI and text generation tasks.

On this page, you’ll find the node parameters for the Cohere Chat Model node, and links to more resources.

-8<- “ snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Model:** Select the model which will generate the completion. n8n dynamically loads available models from the Cohere API. Learn more in the [Cohere model documentation](#).

Node options

- **Sampling Temperature:** Use this option to control the randomness of the sampling process. A higher temperature creates more diverse sampling, but increases the risk of hallucinations.
- **Max Retries:** Enter the maximum number of times to retry a request.

Templates and examples

Related resources

Refer to [Cohere’s API documentation](#) for more information about the service.

-8<- “ snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

DeepSeek Chat Model node

Use the DeepSeek Chat Model node to use DeepSeek’s chat models with conversational [agents](#).

On this page, you’ll find the node parameters for the DeepSeek Chat Model node and links to more resources.

-8<- “ snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

Model

Select the model to use to generate the completion.

n8n dynamically loads models from DeepSeek and you'll only see the models available to your account.

Node options

Use these options to further refine the node's behavior.

Base URL

Enter a URL here to override the default URL for the API.

Frequency Penalty

Use this option to control the chances of the model repeating itself. Higher values reduce the chance of the model repeating itself.

Maximum Number of Tokens

Enter the maximum number of tokens used, which sets the completion length.

Response Format

Choose **Text** or **JSON**. **JSON** ensures the model returns valid JSON.

Presence Penalty

Use this option to control the chances of the model talking about new topics. Higher values increase the chance of the model talking about new topics.

Sampling Temperature

Use this option to control the randomness of the sampling process. A higher temperature creates more diverse sampling, but increases the risk of hallucinations.

Timeout

Enter the maximum request time in milliseconds.

Max Retries

Enter the maximum number of times to retry a request.

Top P

Use this option to set the probability the completion should use. Use a lower value to ignore less probable options.

Templates and examples

Related resources

As DeepSeek is API-compatible with OpenAI, you can refer to [LangChains's OpenAI documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Google Gemini Chat Model node

Use the Google Gemini Chat Model node to use Google’s Gemini chat models with conversational agents.

On this page, you’ll find the node parameters for the Google Gemini Chat Model node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Model:** Select the model to use to generate the completion.

n8n dynamically loads models from the Google Gemini API and you’ll only see the models available to your account.

Node options

- **Maximum Number of Tokens:** Enter the maximum number of tokens used, which sets the completion length.
- **Sampling Temperature:** Use this option to control the randomness of the sampling process. A higher temperature creates more diverse sampling, but increases the risk of hallucinations.
- **Top K:** Enter the number of token choices the model uses to generate the next token.
- **Top P:** Use this option to set the probability the completion should use. Use a lower value to ignore less probable options.
- **Safety Settings:** Gemini supports adjustable safety settings. Refer to Google’s [Gemini API safety settings](#) for information on the available filters and levels.

Limitations

No proxy support

The Google Gemini Chat Model node uses Google's SDK, which doesn't support proxy configuration.

If you need to proxy your connection, as a work around, you can set up a dedicated reverse proxy for Gemini requests and change the **Host** parameter in your [Google Gemini credentials](#) to point to your proxy address:

[Image: Google Gemini credentials proxy configuration]

Templates and examples

Related resources

Refer to [LangChain's Google Gemini documentation](#) for more information about the service.

-8<- “ snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Google Vertex Chat Model node

Use the Google Vertex AI Chat Model node to use Google's Vertex AI chat models with conversational [agents](#).

On this page, you'll find the node parameters for the Google Vertex AI Chat Model node, and links to more resources.

-8<- “ snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Project ID:** Select the project ID from your Google Cloud account to use. n8n dynamically loads projects from the Google Cloud account, but you can also enter it manually.
- **Model Name:** Select the name of the model to use to generate the completion, for example `gemini-1.5-flash-001`, `gemini-1.5-pro-001`, etc. Refer to [Google models](#) for a list of available models.

Node options

- **Maximum Number of Tokens:** Enter the maximum number of tokens used, which sets the completion length.
- **Sampling Temperature:** Use this option to control the randomness of the sampling process. A higher temperature creates more diverse sampling, but increases the risk of hallucinations.
- **Thinking Budget:** Controls reasoning tokens for thinking models. Set to `0` to disable automatic thinking. Set to `-1` for dynamic thinking. Leave empty for auto mode.
- **Top K:** Enter the number of token choices the model uses to generate the next token.
- **Top P:** Use this option to set the probability the completion should

use. Use a lower value to ignore less probable options.

- **Safety Settings:** Gemini supports adjustable safety settings. Refer to Google's [Gemini API safety settings](#) for information on the available filters and levels.

Templates and examples

Related resources

Refer to [LangChain's Google Vertex AI documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Groq Chat Model node

Use the Groq Chat Model node to access Groq's large language models for conversational AI and text generation tasks.

On this page, you'll find the node parameters for the Groq Chat Model node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Model:** Select the model which will generate the completion. n8n dynamically loads available models from the Groq API. Learn more in the [Groq model documentation](#).

Node options

- **Maximum Number of Tokens:** Enter the maximum number of tokens used, which sets the completion length.
- **Sampling Temperature:** Use this option to control the randomness of the sampling process. A higher temperature creates more diverse sampling, but increases the risk of hallucinations.

Templates and examples

Related resources

Refer to [Groq's API documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Mistral Cloud Chat Model node

Use the Mistral Cloud Chat Model node to combine Mistral Cloud's chat models with conversational [agents](#).

On this page, you'll find the node parameters for the Mistral Cloud Chat Model node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Model:** Select the model to use to generate the completion. n8n dynamically loads models from Mistral Cloud and you'll only see the models available to your account.

Node options

- **Maximum Number of Tokens:** Enter the maximum number of tokens used, which sets the completion length.
- **Sampling Temperature:** Use this option to control the randomness of the sampling process. A higher temperature creates more diverse sampling, but increases the risk of hallucinations.
- **Timeout:** Enter the maximum request time in milliseconds.
- **Max Retries:** Enter the maximum number of times to retry a request.
- **Top P:** Use this option to set the probability the completion should use. Use a lower value to ignore less probable options.
- **Enable Safe Mode:** Enable safe mode by injecting a safety prompt at the beginning of the completion. This helps prevent the model from generating offensive content.
- **Random Seed:** Enter a seed to use for random sampling. If set, different calls will generate deterministic results.

Templates and examples

Related resources

Refer to [LangChains's Mistral documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Ollama Chat Model node

The Ollama Chat Model node allows you use local Llama 2 models with conversational [agents](#).

On this page, you'll find the node parameters for the Ollama Chat Model node, and links to more resources.

```
-8<- "snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node parameters

- **Model:** Select the model that generates the completion. Choose from:
 - **Llama2**
 - **Llama2 13B**
 - **Llama2 70B**
 - **Llama2 Uncensored**

Refer to the Ollama [Models Library documentation](#) for more information about available models.

Node options

- **Sampling Temperature:** Use this option to control the randomness of the sampling process. A higher temperature creates more diverse sampling, but increases the risk of hallucinations.
- **Top K:** Enter the number of token choices the model uses to generate the next token.
- **Top P:** Use this option to set the probability the completion should use. Use a lower value to ignore less probable options.

Templates and examples

Related resources

Refer to [LangChains's Ollama Chat Model documentation](#) for more information about the service.

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

```
-8<- "snippets/self-hosting/starter-kits/self-hosted-ai-starter-kit.md"
```

Ollama Chat Model node common issues

Here are some common errors and issues with the [Ollama Chat Model node](#) and steps to resolve or troubleshoot them.

Processing parameters

The Ollama Chat Model node is a [sub-node](#). Sub-nodes behave differently than other nodes when processing multiple items using expressions.

Most nodes, including [root nodes](#), take any number of items as input, process these items, and output the results. You can use expressions to refer to input items, and the node resolves the expression for each item in turn. For example, given an input of five name values, the expression `{{ $json.name }}` resolves to each name in turn.

In sub-nodes, the expression always resolves to the first item. For example, given an input of five name values, the expression `{{ $json.name }}` always resolves to the first name.

Can't connect to a remote Ollama instance

The Ollama Chat Model node supports Bearer token authentication for connecting to remote Ollama instances behind authenticated proxies (such as Open WebUI).

For remote authenticated connections, configure both the remote URL and API key in your Ollama credentials.

Follow the [Ollama credentials instructions](#) for more information.

Can't connect to a local Ollama instance when using Docker

The Ollama Chat Model node connects to a locally hosted Ollama instance using the base URL defined by [Ollama credentials](#). When you run either n8n or Ollama in Docker, you need to configure the network so that n8n can connect to Ollama.

Ollama typically listens for connections on `localhost`, the local network address. In Docker, by default, each container has its own `localhost` which is only accessible from within the container. If either n8n or Ollama are running in containers, they won't be able to connect over `localhost`.

The solution depends on how you're hosting the two components.

If only Ollama is in Docker

If only Ollama is running in Docker, configure Ollama to listen on all interfaces by binding to `0.0.0.0` inside of the container (the official images are already configured this way).

When running the container, [publish the ports](#) with the `-p` flag. By default, Ollama runs on port 11434, so your Docker command should look like this:

```
docker run -d -v ollama:/root/.ollama -p 11434:11434 --name ollama ollama/ollama
```

When configuring [Ollama credentials](#), the `localhost` address should work without a problem (set the **base URL** to `http://localhost:11434`).

If only n8n is in Docker

If only n8n is running in Docker, configure Ollama to listen on all interfaces by binding to `0.0.0.0` on the host.

If you are running n8n in Docker on **Linux**, use the `--add-host` flag to map `host.docker.internal` to `host-gateway` when you start the container. For example:

```
docker run -it --rm --add-host host.docker.internal:host-gateway --  
name n8n -p 5678:5678 -v n8n_data:/home/node/.n8n  
docker.n8n.io/n8nio/n8n
```

If you are using Docker Desktop, this is automatically configured for you.

When configuring [Ollama credentials](#), use `host.docker.internal` as the host address instead of `localhost`. For example, to bind to the default port 11434, you could set the base URL to `http://host.docker.internal:11434`.

If Ollama and n8n are running in separate Docker containers

If both n8n and Ollama are running in Docker in separate containers, you can use Docker networking to connect them.

Configure Ollama to listen on all interfaces by binding to `0.0.0.0` inside of the container (the official images are already configured this way).

When configuring [Ollama credentials](#), use the Ollama container's name as the host address instead of `localhost`. For example, if you call the Ollama container `my-ollama` and it listens on the default port 11434, you would set the base URL to `http://my-ollama:11434`.

If Ollama and n8n are running in the same Docker container

If Ollama and n8n are running in the same Docker container, the `localhost` address doesn't need any special configuration. You can configure Ollama to listen on `localhost` and configure the base URL in the [Ollama credentials in n8n](#) to use `localhost`: `http://localhost:11434`.

Error: connect ECONNREFUSED ::1:11434

This error occurs when your computer has IPv6 enabled, but Ollama is listening to an IPv4 address.

To fix this, change the base URL in your [Ollama credentials](#) to connect to `127.0.0.1`, the IPv4-specific local address, instead of the `localhost` alias that can resolve to either IPv4 or IPv6: `http://127.0.0.1:11434`.

Ollama and HTTP/HTTPS proxies

Ollama doesn't support custom HTTP agents in its configuration. This makes it difficult to use Ollama behind custom HTTP/HTTPS proxies. Depending on your proxy configuration, it might not work at all, despite setting the `HTTP_PROXY` or `HTTPS_PROXY` environment variables.

Refer to [Ollama's FAQ](#) for more information.

OpenAI Chat Model node

Use the OpenAI Chat Model node to use OpenAI's chat models with conversational [agents](#).

On this page, you'll find the node parameters for the OpenAI Chat Model node and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

Model

Select the model to use to generate the completion.

n8n dynamically loads models from OpenAI, and you'll only see the models available to your account.

Use Responses API

OpenAI provides two endpoints for generating output from a model: - **Chat Completions:** The Chat Completions API endpoint generates a model response from a list of messages that comprise a conversation. The API requires the user to handle conversation state manually, for example by adding a [Simple Memory](#) subnode. For new projects, OpenAI recommends to use the Responses API. - **Responses:** The Responses API is an agentic loop, allowing the model to call multiple built-in tools within the span of one API request. It also supports persistent conversations by passing a `conversation_id`.

Toggle to **Use Responses API** if you want the model to generate output using the Responses API. Otherwise, the OpenAI Chat Model node will default to using the Chat Completions API.

Refer to the OpenAI documentation for a [comparison of the Chat Completions and Responses APIs](#).

Built-in Tools

The OpenAI Responses API provides a range of [built-in tools](#) to enrich the model's response. Toggle to **Use Responses API** if you want the model to have access to the following built-in tools:

- **Web Search:** Allows models to search the web for the latest information before generating a response.
- **File Search:** Allow models to search your knowledgebase from

previously uploaded files for relevant information before generating a response. Refer to the [OpenAI documentation](#) for more information.

- **Code Interpreter:** Allows models to write and run Python code in a sandboxed environment.

Node options

Use these options to further refine the node's behavior. The following options are available whether you use the Responses API to generate model output or not.

Frequency Penalty

Use this option to control the chances of the model repeating itself. Higher values reduce the chance of the model repeating itself.

Maximum Number of Tokens

Enter the maximum number of tokens used, which sets the completion length.

Presence Penalty

Use this option to control the chances of the model talking about new topics. Higher values increase the chance of the model talking about new topics.

Sampling Temperature

Use this option to control the randomness of the sampling process. A higher temperature creates more diverse sampling, but increases the risk of hallucinations.

Timeout

Enter the maximum request time in milliseconds.

Max Retries

Enter the maximum number of times to retry a request.

Top P

Use this option to set the probability the completion should use. Use a lower value to ignore less probable options.

Additional node options (Responses API only)

The following, additional options are available when toggling to **Use Responses API**.

Conversation ID

The conversation that this response belongs to. Input items and output items from this response are automatically added to this conversation after this response completes.

Prompt Cache Key

Use this key for caching similar requests to optimize cache hit rates.

Safety Identifier

Apply an identifier to track users who may violate usage policies.

Service Tier

Select the service tier that fits your needs: Auto, Flex, Default, or Priority.

Metadata

A set of key-value pairs for storing structured information. You can attach up to 16 pairs to an object, which is useful for adding custom data that can be used for searching by the API or in the dashboard.

Top Logprobs

Define an integer between 0 and 20 specifying the number of most likely tokens to return at each token position, each with an associated log probability.

Output Format

Choose a response format: Text, JSON Schema, or JSON Object. Use of JSON Schema is recommended, if you want to receive data in JSON format.

Prompt

Configure the prompt filled with a unique ID, its version, and substitutable variables. Prompts are configured through the OpenAI dashboard.

Templates and examples

Related resources

Refer to [LangChains's OpenAI documentation](#) for more information about the service.

Refer to [OpenAI documentation](#) for more information about the parameters.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

OpenAI Chat Model node common issues

Here are some common errors and issues with the [OpenAI Chat Model node](#) and steps to resolve or troubleshoot them.

Processing parameters

The OpenAI Chat Model node is a [sub-node](#). Sub-nodes behave differently than other nodes when processing multiple items using expressions.

Most nodes, including [root nodes](#), take any number of items as input, process these items, and output the results. You can use expressions to refer to input items, and the node resolves the expression for each item in turn. For example, given an input of five name values, the expression {{ \$json.name }} resolves to each name in turn.

In sub-nodes, the expression always resolves to the first item. For example, given an input of five name values, the expression {{ \$json.name }} always resolves to the first name.

-8<- “_snippets/integrations/openai-api-issues.md”

OpenRouter Chat Model node

Use the OpenRouter Chat Model node to use OpenRouter’s chat models with conversational agents.

On this page, you’ll find the node parameters for the OpenRouter Chat Model node and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

Model

Select the model to use to generate the completion.

n8n dynamically loads models from OpenRouter and you'll only see the models available to your account.

Node options

Use these options to further refine the node's behavior.

Frequency Penalty

Use this option to control the chances of the model repeating itself. Higher values reduce the chance of the model repeating itself.

Maximum Number of Tokens

Enter the maximum number of tokens used, which sets the completion length.

Response Format

Choose **Text** or **JSON**. **JSON** ensures the model returns valid JSON.

Presence Penalty

Use this option to control the chances of the model talking about new topics. Higher values increase the chance of the model talking about new topics.

Sampling Temperature

Use this option to control the randomness of the sampling process. A higher temperature creates more diverse sampling, but increases the risk of hallucinations.

Timeout

Enter the maximum request time in milliseconds.

Max Retries

Enter the maximum number of times to retry a request.

Top P

Use this option to set the probability the completion should use. Use a lower value to ignore less probable options.

Templates and examples

Related resources

As OpenRouter is API-compatible with OpenAI, you can refer to [LangChains's OpenAI documentation](#) for more information about the service.

-8<- “ snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Vercel AI Gateway Chat Model node

Use the Vercel AI Gateway Chat Model node to use AI Gateway chat models with conversational agents.

On this page, you'll find the node parameters for the Vercel AI Gateway Chat Model node and links to more resources.

-8<- “ snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

Model

Select the model to use to generate the completion.

n8n dynamically loads models from the AI Gateway and you'll only see the models available to your account.

Node options

Use these options to further refine the node's behavior.

Frequency Penalty

Use this option to control the chance of the model repeating itself. Higher values reduce the chance of the model repeating itself.

Maximum Number of Tokens

Enter the maximum number of tokens used, which sets the completion length.

Response Format

Choose **Text** or **JSON**. **JSON** ensures the model returns valid JSON.

Presence Penalty

Use this option to control the chance of the model talking about new topics. Higher values increase the chance of the model talking about new topics.

Sampling Temperature

Use this option to control the randomness of the sampling process. A higher temperature creates more diverse sampling, but increases the risk of hallucinations.

Timeout

Enter the maximum request time in milliseconds.

Max Retries

Enter the maximum number of times to retry a request.

Top P

Use this option to set the probability the completion should use. Use a lower value to ignore less probable options.

Templates and examples

Related resources

As the Vercel AI Gateway is API-compatible with OpenAI, you can refer to [LangChains's OpenAI documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

xAI Grok Chat Model node

Use the xAI Grok Chat Model node to access xAI Grok's large language models for conversational AI and text generation tasks.

On this page, you'll find the node parameters for the xAI Grok Chat Model node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Model:** Select the model which will generate the completion. n8n dynamically loads available models from the xAI Grok API. Learn more in the [xAI Grok model documentation](#).

Node options

- **Frequency Penalty:** Use this option to control the chances of the model repeating itself. Higher values reduce the chance of the

model repeating itself.

- **Maximum Number of Tokens:** Enter the maximum number of tokens used, which sets the completion length. Most models have a context length of 2048 tokens with the newest models supporting up to 32,768 tokens.
- **Response Format:** Choose **Text** or **JSON**. **JSON** ensures the model returns valid JSON.
- **Presence Penalty:** Use this option to control the chances of the model talking about new topics. Higher values increase the chance of the model talking about new topics.
- **Sampling Temperature:** Use this option to control the randomness of the sampling process. A higher temperature creates more diverse sampling, but increases the risk of hallucinations.
- **Timeout:** Enter the maximum request time in milliseconds.
- **Max Retries:** Enter the maximum number of times to retry a request.
- **Top P:** Use this option to set the probability the completion should use. Use a lower value to ignore less probable options.

Templates and examples

Related resources

Refer to [xAI Grok's API documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Cohere Model node

Use the Cohere Model node to use Cohere's models.

On this page, you'll find the node parameters for the Cohere Model node, and links to more resources.

This node lacks tools support, so it won't work with the [AI Agent](#) node. Instead, connect it with the [Basic LLM Chain](#) node.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node Options

- **Maximum Number of Tokens:** Enter the maximum number of tokens used, which sets the completion length.
- **Sampling Temperature:** Use this option to control the randomness of the sampling process. A higher temperature creates more diverse sampling, but increases the risk of hallucinations.

Templates and examples

Related resources

Refer to [LangChains's Cohere documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Ollama Model node

The Ollama Model node allows you use local Llama 2 models.

On this page, you'll find the node parameters for the Ollama Model node, and links to more resources.

This node lacks tools support, so it won't work with the [AI Agent](#) node. Instead, connect it with the [Basic LLM Chain](#) node.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Model:** Select the model that generates the completion. Choose from:
 - **Llama2**
 - **Llama2 13B**
 - **Llama2 70B**
 - **Llama2 Uncensored**

Refer to the Ollama [Models Library documentation](#) for more information about available models.

Node options

- **Sampling Temperature:** Use this option to control the randomness of the sampling process. A higher temperature creates more diverse sampling, but increases the risk of hallucinations.
- **Top K:** Enter the number of token choices the model uses to generate the next token.
- **Top P:** Use this option to set the probability the completion should use. Use a lower value to ignore less probable options.

Templates and examples

Related resources

Refer to [LangChains's Ollama documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

-8<- “_snippets/self-hosting/starter-kits/self-hosted-ai-starter-kit.md”

Ollama Model node common issues

Here are some common errors and issues with the [Ollama Model node](#) and steps to resolve or troubleshoot them.

Processing parameters

The Ollama Model node is a [sub-node](#). Sub-nodes behave differently than other nodes when processing multiple items using expressions.

Most nodes, including [root nodes](#), take any number of items as input, process these items, and output the results. You can use expressions to refer to input items, and the node resolves the expression for each item in turn. For example, given an input of five name values, the expression `{{ $json.name }}` resolves to each name in turn.

In sub-nodes, the expression always resolves to the first item. For example, given an input of five name values, the expression `{{ $json.name }}` always resolves to the first name.

Can't connect to a remote Ollama instance

The Ollama Model node supports Bearer token authentication for connecting to remote Ollama instances behind authenticated proxies (such as Open WebUI).

For remote authenticated connections, configure both the remote URL and API key in your Ollama credentials.

Follow the [Ollama credentials instructions](#) for more information.

Can't connect to a local Ollama instance when using Docker

The Ollama Model node connects to a locally hosted Ollama instance using the base URL defined by [Ollama credentials](#). When you run either n8n or Ollama in Docker, you need to configure the network so that n8n can connect to Ollama.

Ollama typically listens for connections on `localhost`, the local network address. In Docker, by default, each container has its own `localhost` which is only accessible from within the container. If either n8n or Ollama are running in containers, they won't be able to connect over `localhost`.

The solution depends on how you're hosting the two components.

If only Ollama is in Docker

If only Ollama is running in Docker, configure Ollama to listen on all interfaces by binding to `0.0.0.0` inside of the container (the official images are already configured this way).

When running the container, [publish the ports](#) with the `-p` flag. By default, Ollama runs on port 11434, so your Docker command should look like this:

```
docker run -d -v ollama:/root/.ollama -p 11434:11434 --name ollama ollama/ollama
```

When configuring [Ollama credentials](#), the `localhost` address should work without a problem (set the **base URL** to `http://localhost:11434`).

If only n8n is in Docker

If only n8n is running in Docker, configure Ollama to listen on all interfaces by binding to `0.0.0.0` on the host.

If you are running n8n in Docker on [Linux](#), use the `--add-host` flag to map `host.docker.internal` to `host-gateway` when you start the container. For example:

```
docker run -it --rm --add-host host.docker.internal:host-gateway --name n8n -p 5678:5678 -v n8n_data:/home/node/.n8n docker.n8n.io/n8nio/n8n
```

If you are using Docker Desktop, this is automatically configured for you.

When configuring [Ollama credentials](#), use `host.docker.internal` as the host address instead of `localhost`. For example, to bind to the default port 11434, you could set the base URL to `http://host.docker.internal:11434`.

If Ollama and n8n are running in separate Docker containers

If both n8n and Ollama are running in Docker in separate containers, you can use Docker networking to connect them.

Configure Ollama to listen on all interfaces by binding to `0.0.0.0` inside of the container (the official images are already configured this way).

When configuring [Ollama credentials](#), use the Ollama container's name as the host address instead of `localhost`. For example, if you call the Ollama container `my-ollama` and it listens on the default port 11434, you would set the base URL to `http://my-ollama:11434`.

If Ollama and n8n are running in the same Docker container

If Ollama and n8n are running in the same Docker container, the `localhost` address doesn't need any special configuration. You can configure Ollama to listen on `localhost` and configure the base URL in the [Ollama credentials in n8n](#) to use `localhost: http://localhost:11434`.

Error: connect ECONNREFUSED ::1:11434

This error occurs when your computer has IPv6 enabled, but Ollama is listening to an IPv4 address.

To fix this, change the base URL in your [Ollama credentials](#) to connect to 127.0.0.1, the IPv4-specific local address, instead of the localhost alias that can resolve to either IPv4 or IPv6: <http://127.0.0.1:11434>.

Ollama and HTTP/HTTPS proxies

Ollama doesn't support custom HTTP agents in its configuration. This makes it difficult to use Ollama behind custom HTTP/HTTPS proxies. Depending on your proxy configuration, it might not work at all, despite setting the `HTTP_PROXY` or `HTTPS_PROXY` environment variables.

Refer to [Ollama's FAQ](#) for more information.

Hugging Face Inference Model node

Use the Hugging Face Inference Model node to use Hugging Face's models.

On this page, you'll find the node parameters for the Hugging Face Inference Model node, and links to more resources.

This node lacks tools support, so it won't work with the [AI Agent](#) node. Instead, connect it with the [Basic LLM Chain](#) node.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Model:** Select the model to use to generate the completion.

Node options

- **Custom Inference Endpoint:** Enter a custom inference endpoint URL.
- **Frequency Penalty:** Use this option to control the chances of the model repeating itself. Higher values reduce the chance of the model repeating itself.
- **Maximum Number of Tokens:** Enter the maximum number of tokens used, which sets the completion length.
- **Presence Penalty:** Use this option to control the chances of the model talking about new topics. Higher values increase the chance of the model talking about new topics.
- **Sampling Temperature:** Use this option to control the randomness of the sampling process. A higher temperature creates more diverse sampling, but increases the risk of hallucinations.

- **Top K:** Enter the number of token choices the model uses to generate the next token.
- **Top P:** Use this option to set the probability the completion should use. Use a lower value to ignore less probable options.

Templates and examples

Related resources

Refer to [LangChains's Hugging Face Inference Model documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Chat Memory Manager node

The Chat Memory Manager node manages chat message [memories](#) within your workflows. Use this node to load, insert, and delete chat messages in an in-memory [vector store](#).

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-sub-nodes/chat-memory-manager-purpose.md”

On this page, you'll find a list of operations that the Chat Memory Manager node supports, along with links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Operation Mode:** Choose between **Get Many Messages**, **Insert Messages**, and **Delete Messages** operations.
- **Insert Mode:** Available in **Insert Messages** mode. Choose from:
 - **Insert Messages:** Insert messages alongside existing messages.
 - **Override All Messages:** Replace current memory.
- **Delete Mode:** available in **Delete Messages** mode. Choose from:
 - **Last N:** Delete the last N messages.
 - **All Messages:** Delete messages from memory.
- **Chat Messages:** available in **Insert Messages** mode. Define the chat messages to insert into the memory, including:
 - **Type Name or ID:** Set the message type. Select one of:
 - **AI:** Use this for messages from the AI.
 - **System:** Add a message containing instructions for the AI.
 - **User:** Use this for messages from the user. This message type is sometimes called the 'human' message in other AI tools and guides.
 - **Message:** Enter the message contents.
 - **Hide Message in Chat:** Select whether n8n should display the message to the user in the chat UI (turned off) or not (turned on).

- **Messages Count:** Available in **Delete Messages** mode when you select **Last N**. Enter the number of latest messages to delete.
- **Simplify Output:** Available in **Get Many Messages** mode. Turn on to simplify the output to include only the sender (AI, user, or system) and the text.

Templates and examples

Related resources

Refer to [LangChain's Memory documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Simple Memory node

Use the Simple Memory node to persist chat history in your workflow.

On this page, you'll find a list of operations the Simple Memory node supports, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

Configure these parameters to configure the node:

- **Session Key:** Enter the key to use to store the memory in the workflow data.
- **Context Window Length:** Enter the number of previous interactions to consider for context.

Templates and examples

Related resources

Refer to [LangChain's Buffer Window Memory documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

Simple Memory node common issues

Here are some common errors and issues with the [Simple Memory node](#) and steps to resolve or troubleshoot them.

Single memory instance

If you add more than one Simple Memory node to your workflow, all nodes access the same memory instance by default. Be careful when doing destructive actions that override existing memory contents, such as the override all messages operation in the [Chat Memory Manager node](#). If you want more than one memory instance in your workflow, set different session IDs in different memory nodes.

Managing the Session ID

In most cases, the `sessionId` is automatically retrieved from the **On Chat Message** trigger. But you may run into an error with the phrase `No sessionId`.

If you have this error, first check the output of your Chat trigger to ensure it includes a `sessionId`.

If you're not using the **On Chat Message** trigger, you'll need to manage sessions manually.

For testing purposes, you can use a static key like `my_test_session`. If you use this approach, be sure to set up proper session management before publishing the workflow to avoid potential issues in a live environment.

Motorhead node

Use the Motorhead node to use Motorhead as a [memory server](#).

On this page, you'll find a list of operations the Motorhead node supports, and links to more resources.

Node parameters

- **Session ID:** Enter the ID to use to store the memory in the workflow data.

Node reference

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Templates and examples

Related resources

Refer to [LangChain's Motorhead documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Single memory instance

MongoDB Chat Memory node

Use the MongoDB Chat Memory node to use MongoDB as a [memory](#) server for storing chat history.

On this page, you'll find a list of operations the MongoDB Chat Memory node supports, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Session Key:** Enter the key to use to store the memory in the workflow data.
- **Collection Name:** Enter the name of the collection to store the chat history in. The system will create the collection if it doesn't exist.
- **Database Name:** Enter the name of the database to store the chat history in. If not provided, the database from credentials will be used.
- **Context Window Length:** Enter the number of previous interactions to consider for context.

Related resources

Refer to [LangChain's MongoDB Chat Message History documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Single memory instance

Redis Chat Memory node

Use the Redis Chat Memory node to use Redis as a [memory](#) server.

On this page, you'll find a list of operations the Redis Chat Memory node supports, and links to more resources.

```
-8<- "snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node parameters

- **Session Key:** Enter the key to use to store the memory in the workflow data.
- **Session Time To Live:** Use this parameter to make the session expire after a given number of seconds.
- **Context Window Length:** Enter the number of previous interactions to consider for context.

Templates and examples

Related resources

Refer to [LangChain's Redis Chat Memory documentation](#) for more information about the service.

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

Single memory instance

Postgres Chat Memory node

Use the Postgres Chat Memory node to use Postgres as a [memory](#) server for storing chat history.

On this page, you'll find a list of operations the Postgres Chat Memory node supports, and links to more resources.

```
-8<- "snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node parameters

- **Session Key:** Enter the key to use to store the memory in the workflow data.
- **Table Name:** Enter the name of the table to store the chat history in. The system will create the table if doesn't exist.
- **Context Window Length:** Enter the number of previous interactions to consider for context.

Related resources

Refer to [LangChain's Postgres Chat Message History documentation](#) for more information about the service.

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

Single memory instance

Xata node

Use the Xata node to use Xata as a [memory server](#). On this page, you'll find a list of operations the Xata node supports, and links to more resources.

```
-8<- "snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node parameters

- **Session ID:** Enter the ID to use to store the memory in the workflow data.
- **Context Window Length:** Enter the number of previous interactions to consider for context.

Templates and examples

Related resources

Refer to [LangChain's Xata documentation](#) for more information about the service.

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

Single memory instance

Zep node

Use the Zep node to use Zep as a [memory server](#).

On this page, you'll find a list of operations the Zep node supports, and links to more resources.

```
-8<- "snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node parameters

- **Session ID:** Enter the ID to use to store the memory in the workflow data.

Templates and examples

Related resources

Refer to [LangChain's Zep documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Single memory instance

Auto-fixing Output Parser node

The Auto-fixing Output Parser node wraps another output parser. If the first one fails, it calls out to another LLM to fix any errors.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Templates and examples

Related resources

Refer to [LangChain's output parser documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Item List Output Parser node

Use the Item List Output Parser node to return a list of items with a specific length and separator.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node options

- **Number of Items:** Enter the maximum items to return. Set to -1 for unlimited items.
- **Separator:** Select the separator used to split the results into separate items. Defaults to a new line.

Templates and examples

Related resources

Refer to [LangChain's output parser documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Structured Output Parser node

Use the Structured Output Parser node to return fields based on a JSON Schema.

On this page, you'll find the node parameters for the Structured Output Parser node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Schema Type:** Define the output structure and validation. You have two options to provide the schema:

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-sub-nodes/schema-type-structuring.md”

Templates and examples

Related resources

Refer to [LangChain's output parser documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Common issues

For common questions or issues and suggested solutions, refer to [Common issues](#).

Structured Output Parser node common issues

Here are some common errors and issues with the [Structured Output Parser node](#) and steps to resolve or troubleshoot them.

Processing parameters

The Structured Output Parser node is a [sub-node](#). Sub-nodes behave differently than other nodes when processing multiple items using expressions.

Most nodes, including [root nodes](#), take any number of items as input, process these items, and output the results. You can use expressions to refer to input items, and the node resolves the expression for each item in turn. For example, given an input of five name values, the expression `{{ $json.name }}` resolves to each name in turn.

In sub-nodes, the expression always resolves to the first item. For example, given an input of five name values, the expression `{{ $json.name }}` always resolves to the first name.

Adding the structured output parser node to AI nodes

You can attach output parser nodes to select [AI root nodes](#).

To add the Structured Output Parser to a node, enable the **Require Specific Output Format** option in the AI root node you wish to format. Once the option is enabled, a new **output parser** attachment point is displayed. Click the **output parser** attachment point to add the Structured Output Parser node to the node.

Using the structured output parser to format intermediary steps

The Structured Output Parser node structures the final output from AI agents. It's not intended to structure intermediary output to pass to other AI tools or stages.

To request a specific format for intermediary output, include the response structure in the **System Message** for the **AI Agent**. The message can include either a schema or example response for the agent to use as a template for its results.

Structuring output from agents

Structured output parsing is often not reliable when working with [agents](#).

If your workflow uses agents, n8n recommends using a separate [LLM-chain](#) to receive the data from the agent and parse it. This leads to better, more consistent results than parsing directly in the agent workflow.

Contextual Compression Retriever node

The Contextual Compression Retriever node improves the answers returned from [vector store](#) document similarity searches by taking into account the context from the query.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Templates and examples

Related resources

Refer to [LangChain’s contextual compression retriever documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

MultiQuery Retriever node

The MultiQuery Retriever node automates the process of prompt tuning by using an LLM to generate multiple queries from different perspectives for a given user input query.

On this page, you’ll find the node parameters for the MultiQuery Retriever node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node options

- **Query Count:** Enter how many different versions of the query to generate.

Templates and examples

Related resources

Refer to [LangChain’s retriever conceptual documentation](#) and [LangChain’s multiquery retriever API documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Vector Store Retriever node

Use the Vector Store Retriever node to retrieve documents from a [vector store](#).

On this page, you'll find the node parameters for the Vector Store Retriever node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Limit:** Enter the maximum number of results to return.

Templates and examples

Related resources

Refer to [LangChain's vector store retriever documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Workflow Retriever node

Use the Workflow Retriever node to retrieve data from an n8n workflow for use in a Retrieval QA Chain or another Retriever node.

On this page, you'll find the node parameters for the Workflow Retriever node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

Source

Tell n8n which workflow to call. You can choose either:

- **Database** and enter a workflow ID.
- **Parameter** and copy in a complete [workflow JSON](#).

Workflow values

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-sub-nodes/workflow-values.md”

Templates and examples

Related resources

Refer to [LangChain's general retriever documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Character Text Splitter node

Use the Character Text Splitter node to split document data based on characters.

On this page, you'll find the node parameters for the Character Text Splitter node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

- **Separator:** Select the separator used to split the document into separate items.
- **Chunk Size:** Enter the number of characters in each chunk.
- **Chunk Overlap:** Enter how much overlap to have between chunks.

Templates and examples

Related resources

Refer to [LangChain's text splitter documentation](#) and [LangChain's API documentation for character text splitting](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Recursive Character Text Splitter node

The Recursive Character Text Splitter node splits document data recursively to keep all paragraphs, sentences then words together as long as possible.

On this page, you'll find the node parameters for the Recursive Character Text Splitter node, and links to more resources.

```
-8<- "snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node parameters

- **Chunk Size:** Enter the number of characters in each chunk.
- **Chunk Overlap:** Enter how much overlap to have between chunks.

Templates and examples

Related resources

Refer to [LangChain's text splitter documentation](#) and [LangChain's recursively split by character documentation](#) for more information about the service.

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

Token Splitter node

The Token Splitter node splits a raw text string by first converting the text into BPE tokens, then splits these tokens into chunks and converts the tokens within a single chunk back into text.

On this page, you'll find the node parameters for the Token Splitter node, and links to more resources.

```
-8<- "snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node parameters

- **Chunk Size:** Enter the number of characters in each chunk.
- **Chunk Overlap:** Enter how much overlap to have between chunks.

Templates and examples

Related resources

Refer to [LangChain's token documentation](#) and [LangChain's text splitter documentation](#) for more information about the service.

```
-8<- "snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

AI Agent Tool node

The AI Agent Tool node allows a root-level [agent](#) in your workflow to call other agents as tools to simplify multi-agent orchestration.

The [primary agent](#) can supervise and delegate work to AI Agent Tool nodes that specialize in different tasks and knowledge. This allows you to use multiple agents in a single workflow without the complexity of managing context and variables that sub-workflows require. You can nest AI Agent Tool nodes into multiple layers for more complex multi-tiered use cases.

On this page, you'll find the node parameters for the AI Agent Tool node, and links to more resources.

-8<- “ snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

Configure the AI Agent Tool node using these parameters:

- **Description:** Give a description to the LLM of this agent's purpose and scope of responsibility. A good, specific description tells the parent agent when to delegate tasks to this agent for processing.
- **Prompt (User Message):** The prompt to the LLM explaining what actions to perform and what information to return.
- **Require Specific Output Format:** Whether you want the node to require a specific output format. When turned on, n8n prompts you to connect one of the output parsers [described on the main agent page](#).
- **Enable Fallback Model:** Whether to enable a fallback model. When enabled, n8n prompts you to connect a backup chat model to use in case the primary model fails or isn't available.

Node options

Refine the AI Agent Tool node's behavior using these options:

- **System Message:** A message to send to the agent before the conversation starts.
- **Max Iterations:** The maximum number of times the model should run to generate a response before stopping.
- **Return Intermediate Steps:** Whether to include intermediate steps the agent took in the final output.
- **Automatically Passthrough Binary Images:** Whether binary images should be automatically passed through to the agent as image type messages.
- **Batch Processing:** Whether to enable the following batch processing options for rate limiting:
 - **Batch Size:** The number of items to process in parallel. This helps with rate limiting but may impact the log output ordering.
 - **Delay Between Batches:** The number of milliseconds to wait between batches.

Templates and examples

Dynamic parameters for tools with \$fromAI()

To learn how to dynamically populate parameters for app node tools, refer to [Let AI specify tool parameters with \\$fromAI\(\)](#).

Calculator node

The Calculator node is a [tool](#) that allows an [agent](#) to run mathematical calculations.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Templates and examples

Related resources

-8<- “_snippets/integrations/builtin/cluster-nodes/tools-link.md”

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Custom Code Tool node

Use the Custom Code Tool node to write code that an [agent](#) can run.

On this page, you'll find the node parameters for the Custom Code Tool node and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

Description

Give your custom code a description. This tells the agent when to use this tool. For example:

Call this tool to get a random color. The input should be a string with comma separated names of colors to exclude.

Language

You can use JavaScript or Python.

JavaScript / Python box

Write the code here.

You can access the tool input using `query`. For example, to take the input string and lowercase it:

```
let myString = query;
return myString.toLowerCase();
```

Templates and examples

Related resources

-8<- “_snippets/integrations/builtin/cluster-nodes/tools-link.md”
-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

MCP Client Tool node

The MCP Client Tool node is a [Model Context Protocol \(MCP\)](#) 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.

Node parameters

Configure the node with the following parameters.

- **SSE Endpoint:** The SSE endpoint for the MCP server you want to connect to.
- **Authentication:** The authentication method for authentication to your MCP server. The MCP tool supports [bearer](#), generic [header](#), and [OAuth2](#) authentication. Select **None** to attempt to connect without authentication.
- **Tools to Include:** Choose which tools you want to expose to the AI Agent:
 - **All:** Expose all the tools given by the MCP server.
 - **Selected:** Activates a **Tools to Include** parameter where you can select the tools you want to expose to the AI Agent.
 - **All Except:** Activates a **Tools to Exclude** parameter where you can select the tools you want to avoid sharing with the AI Agent. The AI Agent will have access to all MCP server's tools that aren't selected.

Templates and examples

Related resources

n8n also has an [MCP Server Trigger](#) node that allows you to expose n8n tools to external AI Agents.

Refer to the [MCP documentation](#) and [MCP specification](#) for more details about the protocol, servers, and clients.

```
-8<- "_snippets/integrations/builtin/cluster-nodes/tools-link.md"
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

SearXNG Tool node

The SearXNG Tool node allows you to integrate search capabilities into your workflows using SearXNG. SearXNG aggregates results from multiple search engines without tracking you.

On this page, you'll find the node options for the SearXNG Tool node, and links to more resources.

```
-8<- "_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node Options

- **Number of Results:** The number of results to retrieve. The default is 10.
- **Page Number:** The page number of the search results to retrieve. The default is 1.
- **Language:** A two-letter [language code](#) to filter search results by language. For example: en for English, fr for French. The default is en.
- **Safe Search:** Enables or disables filtering explicit content in the search results. Can be None, Moderate, or Strict. The default is None.

Running a SearXNG instance

This node requires running the SearXNG service on the same network as your n8n instance. Ensure your n8n instance has network access to the SearXNG service.

This node requires results in JSON format, which isn't enabled in the default SearXNG configuration. To enable JSON output, add json to the `search.formats` section of your SearXNG instance's `settings.yml` file:

```
search:
  # options available for formats: [html, csv, json, rss]
  formats:
    - html
    - json
```

If the `formats` section isn't there, add it. The exact location of the `settings.yml` file depends on how you installed SearXNG. You can find more by visiting the [SearXNG configuration documentation](#).

The quality and availability of search results depend on the configuration and health of the SearXNG instance you use.

Templates and examples

Related resources

Refer to [SearXNG's documentation](#) for more information about the service. You can also view [LangChain's documentation on their SearXNG integration](#).

-8<- “[_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md](#)”

SerpApi (Google Search) node

The SerpAPI node allows an [agent](#) in your workflow to call Google's Search API.

On this page, you'll find the node parameters for the SerpAPI node, and links to more resources.

-8<- “[_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md](#)”

Node options

- **Country:** Enter the country code you'd like to use. Refer to [Google GL Parameter: Supported Google Countries](#) for supported countries and country codes.
- **Device:** Select the device to use to get the search results.
- **Explicit Array:** Choose whether to force SerpApi to fetch the Google results even if a cached version is already present (turned on) or not (turned off).
- **Google Domain:** Enter the Google Domain to use. Refer to [Supported Google Domains](#) for supported domains.
- **Language:** Enter the language code you'd like to use. Refer to [Google HL Parameter: Supported Google Languages](#) for supported languages and language codes.

Templates and examples

Related resources

Refer to [Serp's documentation](#) for more information about the service. You can also view [LangChain's documentation on their Serp integration](#).

-8<- “[_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md](#)”

Think Tool node

The Think Tool node is a [tool](#) that invites an [agent](#) to do some thinking before answering. Essentially, it allows the agent to reflect on the question by talking to itself before providing an answer. In particular, this is useful for complex queries that require a thoughtful response.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Templates and examples

Prompting example

As an example, say you ask the agent:

Jack is looking at Anne. Anne is looking at George. Jack is married, George is not, and we don't know if Anne is married. Is a married person looking at an unmarried person?

The agent will do its thinking with the tool:

We have the following information: 1. Jack is looking at Anne. 2. Anne is looking at George. 3. Jack is married. 4. George is not married. 5. We don't know if Anne is married or not.

We need to determine if a married person is looking at an unmarried person. Let's consider both cases for Anne:

Case 1: Anne is married.

- If Anne is married, then she (married) is looking at George (unmarried). In this case, a married person (Anne) is looking at an unmarried person (George).*

Case 2: Anne is unmarried.

- If Anne is unmarried, then Jack (married) is looking at Anne (unmarried). In this case, a married person (Jack) is looking at an unmarried person (Anne).*

In both scenarios, we find that a married person is looking at an unmarried person. Therefore, the answer to the question is yes.

and then afterward provide the answer:

Yes, a married person is looking at an unmarried person. In both possible scenarios regarding Anne's marital status, we find that a married individual (either Jack or Anne) is looking at an unmarried individual (either George or Anne).

Related resources

-8<- “_snippets/integrations/builtin/cluster-nodes/tools-link.md”

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Vector Store Question Answer Tool node

The Vector Store Question Answer node is a [tool](#) that allows an [agent](#) to summarize results and answer questions based on chunks from a [vector store](#).

On this page, you'll find the node parameters for the Vector Store Question Answer node, and links to more resources.

-8<- “_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md”

Node parameters

Description of Data

Enter a description of the data in the vector store.

Limit

The maximum number of results to return.

How n8n populates the tool description

n8n uses the node name (select the name to edit) and **Description of Data** parameter to populate the tool description for AI agents using the following format:

Useful for when you need to answer questions about [node name]. Whenever you need information about [Description of Data](#), you should ALWAYS use this. Input should be a fully formed question.

Spaces in the node name are converted to underscores in the tool description.

Related resources

View [example workflows and related content](#) on n8n's website.

-8<- “_snippets/integrations/builtin/cluster-nodes/tools-link.md”

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Wikipedia node

The Wikipedia node is a [tool](#) that allows an [agent](#) to search and return information from Wikipedia.

```
-8<- "_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Templates and examples

Related resources

```
-8<- "_snippets/integrations/builtin/cluster-nodes/tools-link.md"
```

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

Wolfram|Alpha tool node

Use the Wolfram|Alpha tool node to connect your [agents](#) and [chains](#) to Wolfram|Alpha's computational intelligence engine.

```
-8<- "_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Templates and examples

Related resources

Refer to [Wolfram|Alpha's documentation](#) for more information about the service. You can also view [LangChain's documentation on their WolframAlpha Tool](#).

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

Call n8n Workflow Tool node

The Call n8n Workflow Tool node is a [tool](#) that allows an [agent](#) to run another n8n workflow and fetch its output data.

On this page, you'll find the node parameters for the Call n8n Workflow Tool node, and links to more resources.

```
-8<- "_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node parameters

Description

Enter a custom code a description. This tells the agent when to use this tool. For example:

Call this tool to get a random color. The input should be a string with comma separated names of colors to exclude.

Source

Tell n8n which workflow to call. You can choose either:

- **Database** to select the workflow from a list or enter a workflow ID.
- **Define Below** and copy in a complete [workflow JSON](#).

Workflow Inputs

When using **Database** as workflow source, once you choose a sub-workflow (and define the **Workflow Input Schema** in the sub-workflow), you can define the **Workflow Inputs**.

Select the **Refresh** button to pull in the input fields from the sub-workflow.

You can define the workflow input values using any combination of the following options:

- providing fixed values
- using expressions to reference data from the current workflow
- [letting the AI model specify the parameter](#) by selecting the button AI button on the right side of the field
- using the [\\$fromAI\(\) function](#) in expressions to control the way the model fills in data and to mix AI generated input with other custom input

To reference data from the current workflow, drag fields from the input panel to the field with the Expressions mode selected.

To get started with the `$fromAI()` function, select the “Let the model define this parameter” button on the right side of the field and then use the **X** on the box to revert to user-defined values. The field will change to an expression field pre-populated with the `$fromAI()` expression. From here, you can customize the expression to add other static or dynamic content, or tweak the `$fromAI()` function parameters.

Templates and examples

Related resources

-8<- “_snippets/integrations/builtin/cluster-nodes/tools-link.md”

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Reranker Cohere

The Reranker Cohere node allows you to [rerank](#) the resulting chunks from a [vector store](#). You can connect this node to a vector store.

The reranker reorders the list of documents retrieved from a vector store for a given query in order of descending relevance.

On this page, you'll find the node parameters for the Reranker Cohere node, and links to more resources.

```
-8<- "_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node parameters

Model

Choose the reranking model to use. You can find out more about the available models in [Cohere's model documentation](#).

Templates and examples

Related resources

```
-8<- "_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md"
```

Model Selector

The Model Selector node dynamically selects one of the connected language models during workflow execution based on a set of defined conditions. This enables implementing fallback mechanisms for error handling or choosing the optimal model for specific tasks.

This page covers node parameters for the Model Selector node and includes links to related resources.

```
-8<- "_snippets/integrations/builtin/cluster-nodes/sub-node-expression-resolution.md"
```

Node parameters

Number of Inputs

Specifies the number of input connections available for attaching language models.

Rules

Each rule defines the model to use when specific conditions match.

The Model Selector node evaluates rules sequentially, starting from the first input, and stops evaluation as soon as it finds a match. This means that if multiple rules would match, n8n will only use the model defined by the first matching rule.

Templates and examples

Related resources

-8<- “[snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md](#)”

Credentials library

This section contains step-by-step information about authenticating the different nodes in n8n.

To learn more about creating, managing, and sharing credentials, refer to [Manage credentials](#).

Action Network credentials

You can use these credentials to authenticate the following nodes:

- [Action Network](#)

Supported authentication methods

- API key

Related resources

Refer to [Action Network’s API documentation](#) for more information about working with the service.

Using API key

To configure this credential, you’ll need an [Action Network](#) account with [API key access enabled](#) and:

- An **API Key**

To get an API key:

1. Log in to your Action Network account.
2. From the **Start Organizing** menu, select **Details > API & Sync**.
3. Select the list you want to generate an API key for.
4. Generate an API key for that list.
5. Copy the **API Key** and enter it in your n8n credential.

Refer to the [Action Network API Authentication instructions](#) for more information.

Request API access

Each user account and group on the Action Network has a separate API key to access that user or group's data.

You must explicitly request API access from Action Network, which you can do in one of two ways:

1. If you're already a paying customer, [contact them](#) to request partner access. Partner access includes API key access.
 2. If you're a developer, [request a developer account](#). Once your account request is granted, you'll have API key access.
-

ActiveCampaign credentials

You can use these credentials to authenticate the following nodes:

- [ActiveCampaign](#)
- [Active Campaign Trigger](#)

Supported authentication methods

- API key

Related resources

Refer to [ActiveCampaign's API documentation](#) for more information about working with the service.

Using API key

To configure this credential, you'll need an [ActiveCampaign](#) account and:

- An **API URL**
- An **API Key**

To get both and set up the credential:

1. In ActiveCampaign, select **Settings** (the gear cog icon) from the left menu.
2. Select **Developer**.
3. Copy the **API URL** and enter it in your n8n credential.
4. Copy the **API Key** and enter it in your n8n credential.

Refer to [How to obtain your ActiveCampaign API URL and Key](#) for more information or for instructions on resetting your API key.

Acuity Scheduling credentials

You can use these credentials to authenticate the following nodes:

- [Acuity Scheduling Trigger](#)

Prerequisites

Create an [Acuity Scheduling](#) account.

Supported authentication methods

- API key
- OAuth2

Related resources

Refer to [Acuity's API documentation](#) for more information about working with the service.

Using API key

To configure this credential, you'll need:

- A numeric **User ID**
- An **API Key**

Refer to the [Acuity API Quick Start authentication instructions](#) to generate an API key and view your User ID.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you need to set this up from scratch, complete the [Acuity OAuth2 Account Registration page](#). Use the **Client ID** and **Client Secret** provided from that registration.

Adalo credentials

You can use these credentials to authenticate the following nodes:

- [Adalo](#)

Supported authentication methods

- API key

Related resources

Refer to [Adalo's API collections documentation](#) for more information about working with the service.

Using API key

To configure this credential, you'll need an [Adalo](#) account and:

- An **API Key**
- An **App ID**

To get these, create an Adalo app:

1. From the app dropdown in the top navigation, select **CREATE NEW APP**.
2. Select the App Layout type that makes sense for you and select **Next**.
 - If you're new to using the product, Adalo recommend using **Mobile Only**.
3. Select a template to get started with or select **Blank**, then select **Next**.
4. Enter an **App Name**, like n8n integration.
5. If applicable, select the **Team** for the app.
6. Select branding colors.
7. Select **Create**. The app editor opens.
8. In the left menu, select **Settings** (the gear cog icon).
9. Select **App Access**.
10. In the **API Key** section, select **Generate Key**.
 - If you don't have the correct plan level, you'll see a prompt to upgrade instead.
11. Copy the key and enter it as the **API Key** in your n8n credential.
12. The URL includes the **App ID** after <https://app.adalo.com/apps/>.
For example, if the URL for your app is
<https://app.adalo.com/apps/b78bdfcf-48dc-4550-a474-dd52c19fc371> is the App ID. Copy this value and enter it in your n8n credential.

Refer to [Creating an app](#) for more information on creating apps in Adalo. Refer to [The Adalo API](#) for more information on generating API keys.

Affinity credentials

You can use these credentials to authenticate the following nodes:

- [Affinity](#)
- [Affinity Trigger](#)

Prerequisites

Create an [Affinity](#) account at the Scale, Advanced, or Enterprise subscription tiers.

Supported authentication methods

- API key

Related resources

Refer to [Affinity's API documentation](#) for more information about working with the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Refer to [How to obtain your Affinity API key documentation](#) to get your API key.
-

Agile CRM credentials

You can use these credentials to authenticate the following nodes:

- [Agile CRM](#)

Prerequisites

Create an [Agile CRM](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Agile CRM's API documentation](#) for more information about working with the service.

Using API key

To configure this credential, you'll need:

- An **Email Address** registered with AgileCRM
 - A **REST API Key**: Access your Agile CRM API key through **Admin Settings > Developers & API > REST API key**.
 - An Agile CRM **Subdomain** (for example, n8n)
-

Airtable credentials

You can use these credentials to authenticate the following nodes:

- [Airtable](#)
- [Airtable Trigger](#)

Prerequisites

Create an [Airtable](#) account.

Supported authentication methods

- Personal Access Token (PAT)
- OAuth2

Related resources

Refer to [Airtable's API documentation](#) for more information about the service.

Using personal access token

To configure this credential, you'll need:

- A Personal **Access Token** (PAT)

To create your PAT:

1. Go to the Airtable Builder Hub [Personal access tokens](#) page.
2. Select **+ Create new token**. Airtable opens the **Create personal access token** page.
3. Enter a **Name** for your token, like n8n credential.
4. Add **Scopes** to your token. Refer to Airtable's [Scopes](#) guide for more information. n8n recommends using these scopes:
 - data.records:read
 - data.records:write
 - schema.bases:read
5. Select the **Access** for your token. Choose from a single base, multiple bases (even bases from different workspaces), all of the current and future bases in a workspace you own, or all of the bases from any workspace that you own including bases/workspace added in the future.
6. Select **Create token**.
7. Airtable opens a modal with your token displayed. Copy this token and enter it in your n8n credential as the **Access Token**.

Refer to Airtable's [Find/create PATs documentation](#) for more information.

Using OAuth2

To configure this credential, you'll need:

- An **OAuth Redirect URL**
- A **Client ID**
- A **Client Secret**

To generate all this information, register a new Airtable integration:

1. Open your Airtable Builder Hub [OAuth integrations](#) page.
2. Select the **Register new OAuth integration** button.

3. Enter a name for your OAuth integration.
4. Copy the **OAuth Redirect URL** from your n8n credential.
5. Paste that redirect URL in Airtable as the **OAuth redirect URL**.
6. Select **Register integration**.
7. On the following page, copy the **Client ID** from Airtable and paste it into the **Client ID** in your n8n credential.
8. In Airtable, select **Generate client secret**.
9. Copy the client secret and paste it into the **Client Secret** in your n8n credential.
10. Select the following scopes in Airtable:
 - data.records:read
 - data.records:write
 - schema.bases:read
11. Select **Save changes** in Airtable.
12. In your n8n credential, select the **Connect my account**. A **Grant access** modal opens.
13. Follow the instructions and select the base you want to work on (or all bases).
14. Select **Grant access** to complete the connection.

Refer to the [Airtable Register a new integration documentation](#) for steps on registering a new Oauth integration.

Airtop credentials

You can use these credentials to authenticate the following nodes:

- [Airtop](#)

Prerequisites

Create an [Airtop](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Airtop's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need an [Airtop](#) account and an **API key**. To generate a new key:

1. Log in to the [Airtop Portal](#).
2. Go to [API Keys](#).
3. Select the **+ Create new key** button.
4. Enter a name for the API key.
5. Select the generated key to copy the key.

6. Enter this as the **API Key** in your n8n credential.

Refer to [Airtop's Support](#) for assistance if you have any issues creating your API key.

AlienVault credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create an [AlienVault](#) account.

Supported authentication methods

- API key

Related resources

Refer to [AlienVault's documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n's website.

Using API key

To configure this credential, you'll need:

- An **OTX Key**: Once you have an AlienVault account, the **OTX Key** displays in your **Settings**.
-

AMQP credentials

You can use these credentials to authenticate the following nodes:

- [AMQP Sender](#)
- [AMQP Trigger](#)

Prerequisites

Install an AMQP 1.0-compatible message broker like [ActiveMQ](#). Refer to [AMQP Products](#) for a list of options.

Supported authentication methods

- AMQP connection

Related resources

Advanced Message Queuing Protocol (AMQP) is an open standard application layer protocol for message-oriented middleware. The defining features of AMQP are message orientation, queuing, routing, reliability and security. Refer to the [OASIS AMQP Version 1.0 Standard](#) for more information.

Refer to your provider's documentation for more information about the service. Refer to [ActiveMQ's API documentation](#) as one example.

Using AMQP connection

To configure this credential, you'll need:

- A **Hostname**: Enter the hostname of your AMQP message broker.
- A **Port**: Enter the port number the connection should use.
- A **User**: Enter the name of the user to establish the connection as.
 - For example, the default username in ActiveMQ is `admin`.
- A **Password**: Enter the user's password.
 - For example, the default password in ActiveMQ is `admin`.
- *Optional: Transport Type*: Enter either `tcp` or `tls`.

Refer to your provider's documentation for more detailed instructions.

Anthropic credentials

You can use these credentials to authenticate the following nodes:

- [Anthropic](#)
- [Anthropic Chat Model](#)

Supported authentication methods

- API key

Related resources

Refer to [Anthropic's documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Using API key

To configure this credential, you'll need an [Anthropic Console account](#) with access to Claude.

Then:

1. In the Anthropic Console, open **Settings > API Keys**.
2. Select **+ Create Key**.
3. Give your key a **Name**, like n8n-integration.
4. Select **Copy Key** to copy the key.
5. Enter this as the **API Key** in your n8n credential.

Refer to Anthropic's [Intro to Claude](#) and [Quickstart](#) for more information.

APITemplate.io credentials

You can use these credentials to authenticate the following nodes:

- [APITemplate.io](#)

Prerequisites

Create an [APITemplate.io](#) account.

Supported authentication methods

- API key

Related resources

Refer to [APITemplate.io's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Once you've created an APITemplate.io account, go to [API Integration](#) to copy the **API Key**.
-

Asana credentials

You can use these credentials to authenticate the following nodes:

- [Asana](#)
- [Asana Trigger](#)

Supported authentication methods

- Access token
- OAuth2

Related resources

Refer to [Asana's Developer Guides](#) for more information about working with the service.

Using Access token

To configure this credential, you'll need an [Asana](#) account and:

- A Personal **Access Token** (PAT)

To get your PAT:

1. Open the Asana [developer console](#).
2. In the **Personal access tokens** section, select **Create new token**.
3. Enter a **Token name**, like n8n integration.
4. Check the box to agree to the [Asana API terms](#).
5. Select **Create token**.
6. Copy the token and enter it as the **Access Token** in your n8n credential.

Refer to the [Asana Quick start guide](#) for more information.

Using OAuth2

To configure this credential, you'll need an [Asana](#) account.

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you're [self-hosting](#) n8n, you'll need to register an application to set up OAuth:

1. Open the Asana [developer console](#).
2. In the **My apps** section, select **Create new app**.
3. Enter an **App name** for your application, like n8n integration.
4. Select a purpose for your app.
5. Check the box to agree to the [Asana API terms](#).
6. Select **Create app**. The page opens to the app's **Basic Information**.
7. Select **OAuth** from the left menu.
8. In n8n, copy the **OAuth Redirect URL**.
9. In Asana, select **Add redirect URL** and enter the URL you copied from n8n.
10. Copy the **Client ID** from Asana and enter it in your n8n credential.
11. Copy the **Client Secret** from Asana and enter it in your n8n credential.

Refer to the [Asana OAuth register an application documentation](#) for more information.

Auth0 Management credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create an [Auth0](#) account.

Supported authentication methods

- API client secret

Related resources

Refer to [Auth0 Management's documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n's website.

Using API client secret

To configure this credential, you'll need:

- An Auth0 **Domain**
- A **Client ID**
- A **Client Secret**

Refer to the [Auth0 Management API Get Access Tokens documentation](#) for instructions on obtaining the Client ID and Client Secret from the application's **Settings** tab.

Autopilot credentials

You can use these credentials to authenticate the following nodes:

- [Autopilot](#)
- [Autopilot Trigger](#)

Prerequisites

Create an [Autopilot](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Autopilot's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Generate an API key in **Settings > Autopilot API**. Refer to [Autopilot API authentication](#) for more information.
-

AWS credentials

AWS (IAM) credentials

You can use these credentials to authenticate the following nodes:

- [AWS Bedrock Chat Model](#)
- [AWS Certificate Manager](#)
- [AWS Cognito](#)
- [AWS Comprehend](#)
- [AWS DynamoDB](#)
- [AWS Elastic Load Balancing](#)
- [AWS IAM](#)
- [AWS Lambda](#)
- [AWS Rekognition](#)
- [AWS S3](#)
- [AWS SES](#)
- [AWS SNS](#)
- [AWS SNS Trigger](#)
- [AWS SQS](#)
- [AWS Textract](#)
- [AWS Transcribe](#)
- [Embeddings AWS Bedrock](#)

Supported authentication methods

- API access key

Related resources

Refer to [AWS's Identity and Access Management documentation](#) for more information about the service.

Using API access key

To configure this credential, you'll need an [AWS](#) account and:

- Your **AWS Region**
- The **Access Key ID**: Generated when you create an access key.
- The **Secret Access Key**: Generated when you create an access key.

To create an access key and set up the credential:

1. In your n8n credential, select your **AWS Region**.
2. Log in to the [IAM console](#).
3. In the navigation bar on the upper right, select your user name

and then select **Security credentials**.

4. In the **Access keys** section, select **Create access key**.
5. On the **Access key best practices & alternatives page**, choose your use case. If it doesn't prompt you to create an access key, select **Other**.
6. Select **Next**.
7. Set a **description** tag value for the access key to make it easier to identify, for example n8n integration.
8. Select **Create access key**.
9. Reveal the **Access Key ID** and **Secret Access Key** and enter them in n8n.
10. To use a **Temporary security credential**, turn that option on and add a **Session token**. Refer to the [AWS Temporary security credential documentation](#) for more information on working with temporary security credentials.
11. If you use [Amazon Virtual Private Cloud \(VPC\)](#) to host n8n, you can establish a connection between your VPC and some apps. Use **Custom Endpoints** to enter relevant custom endpoint(s) for this connection. This setup works with these apps:
 - Rekognition
 - Lambda
 - SNS
 - SES
 - SQS
 - S3

You can also generate access keys through the AWS CLI and AWS API. Refer to the [AWS Managing Access Keys documentation](#) for instructions on generating access keys using these methods.

AWS (Assume Role) credentials

You can use these credentials to authenticate the following nodes with enhanced security through IAM role assumption:

- [AWS Certificate Manager](#)
- [AWS Cognito](#)
- [AWS Comprehend](#)
- [AWS DynamoDB](#)
- [AWS Elastic Load Balancing](#)
- [AWS Rekognition](#)
- [AWS S3](#)
- [AWS SES](#)
- [AWS SQS](#)
- [AWS Textract](#)
- [AWS Transcribe](#)

Supported authentication methods

- Role Assumption

Related resources

Refer to [AWS's IAM Role documentation](#) and [STS AssumeRole documentation](#) for more information about role assumption.

Understanding AWS Role Assumption

AWS Role Assumption allows you to securely access AWS resources by temporarily assuming an IAM role, rather than using long-lived access keys. This follows AWS security best practices and enables:

- **Cross-account access:** Access resources in different AWS accounts
- **Enhanced security:** Use temporary credentials that automatically expire
- **Principle of least privilege:** Grant only the permissions needed for specific tasks
- **Audit trail:** Better tracking of who accessed what resources

Setting up AWS Assume Role credentials

To configure this credential, you'll need:

Required Parameters

- **Region:** The AWS region in which to call the STS service to assume the role.
- **Role ARN:** The Amazon Resource Name (ARN) of the IAM role you want to assume. It has the format `arn:aws:iam::123456789012:role/MyRole`. This role must have a trust policy that allows your credentials to assume it.
- **External ID:** A unique identifier required by the role's trust policy to prevent the "confused deputy" problem. This should be a secret value that you generate and configure in both the role's trust policy and this credential. Treat this value as sensitive. Don't share it with other n8n users you don't trust.
- **Role Session Name:** A name for the assumed role session (used for auditing). Default values is `n8n-session`. This value appears in AWS CloudTrail logs so you can identify the session.

STS credentials (Choose one method)

You have two options for providing credentials to make the STS AssumeRole call:

Option 1: Use system credentials (recommended for server deployments)

Enable this option if your n8n server has AWS credentials configured through:

- Environment variables
(`AWS_ACCESS_KEY_ID`, `AWS_SECRET_ACCESS_KEY`, `AWS_SESSION_TOKEN`)
- EC2 instance profile
- ECS task role
- EKS pod identity

This option requires your n8n administrator to enable system credentials access by setting environment variable `N8N_AWS_SYSTEM_CREDENTIALS_ACCESS_ENABLED` to `true`

Option 2: Manual STS Credentials

If system credentials aren't available, provide these manually:

- **STS Access Key ID:** Access Key ID for an IAM user or role that has permission to assume the target role.
- **STS Secret Access Key:** Secret Access Key corresponding to the STS Access Key ID.
- **STS Session Token** (optional): Session token if using temporary credentials for the STS call.

Optional Parameters

- **Custom Endpoints:** If using Amazon VPC, you can specify custom endpoints for AWS services:
 - Rekognition Endpoint
 - Lambda Endpoint
 - SNS Endpoint
 - SES Endpoint
 - SQS Endpoint
 - S3 Endpoint
 - SSM Endpoint

Setup Steps

1. Create the IAM Role in the target AWS account.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::SOURCE-ACCOUNT:root"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "sts:ExternalId": "your-unique-external-id"
        }
      }
    }
  ]
}
```

2. Configure the credential in n8n.

- Select your **AWS Region**
- Enter the **Role ARN** of the role you created
- Set a unique **External ID** (same as in the trust policy)
- Choose your **STS credentials method**
- Enter the **Role Session Name** (or use default)

3. **Test the credential** using the built-in test function to verify the role assumption works.

Security Best Practices

- Use unique External IDs for each credential to prevent unauthorized access.
- Rotate the STS credentials used for role assumption.
- Apply the principle of least privilege to both the assuming

credentials and the target role.

Azure OpenAI credentials

You can use these credentials to authenticate the following nodes:

- [Chat Azure OpenAI](#)
- [Embeddings Azure OpenAI](#)

Prerequisites

- Create an [Azure](#) subscription.
- Access to Azure OpenAI within that subscription. You may need to [request access](#) if your organization doesn't yet have it.

Supported authentication methods

- API key
- Azure Entra ID (OAuth2)

Related resources

Refer to [Azure OpenAI's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- A **Resource Name**: the **Name** you give the resource
- An **API key**: **Key 1** works well. This can be accessed before deployment in **Keys and Endpoint**.
- The **API Version** the credentials should use. See the [Azure OpenAI API preview lifecycle documentation](#) for more information about API versioning in Azure OpenAI.

To get the information above, [create and deploy an Azure OpenAI Service resource](#).

Using Azure Entra ID (OAuth2)

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

For self-hosted users, there are two main steps to configure OAuth2 from scratch:

1. [Register an application](#) with the Microsoft Identity Platform.
2. [Generate a client secret](#) for that application.

Follow the detailed instructions for each step below. For more detail on the Microsoft OAuth2 web flow, refer to [Microsoft authentication and authorization basics](#).

Register an application

Register an application with the Microsoft Identity Platform:

1. Open the [Microsoft Application Registration Portal](#).
2. Select **Register an application**.
3. Enter a **Name** for your app.
4. In **Supported account types**, select **Accounts in any organizational directory (Any Azure AD directory - Multi-tenant) and personal Microsoft accounts (for example, Skype, Xbox)**.
5. In **Register an application**:
 1. Copy the **OAuth Callback URL** from your n8n credential.
 2. Paste it into the **Redirect URI (optional)** field.
 3. Select **Select a platform > Web**.
6. Select **Register** to finish creating your application.
7. Copy the **Application (client) ID** and paste it into n8n as the **Client ID**.

Refer to [Register an application with the Microsoft Identity Platform](#) for more information.

Generate a client secret

With your application created, generate a client secret for it:

1. On your Microsoft application page, select **Certificates & secrets** in the left navigation.
2. In **Client secrets**, select **+ New client secret**.
3. Enter a **Description** for your client secret, such as n8n credential.
4. Select **Add**.
5. Copy the **Secret** in the **Value** column.
6. Paste it into n8n as the **Client Secret**.
7. Select **Connect my account** in n8n to finish setting up the connection.
8. Log in to your Microsoft account and allow the app to access your info.

Refer to Microsoft's [Add credentials](#) for more information on adding a client secret.

Setting custom scopes

Azure Entra ID credentials use the following scopes by default:

- [openid](#)
- [offline_access](#)
- [AccessReview.ReadWrite.All](#)
- [Directory.ReadWrite.All](#)
- [NetworkAccessPolicy.ReadWrite.All](#)
- [DelegatedAdminRelationship.ReadWrite.All](#)
- [EntitlementManagement.ReadWrite.All](#)
- [User.ReadWrite.All](#)
- [Directory.AccessAsUser.All](#)

- [Sites.FullControl.All](#)
- [GroupMember.ReadWrite.All](#)

To select different scopes for your credentials, enable the **Custom Scopes** slider and edit the **Enabled Scopes** list. Keep in mind that some features may not work as expected with more restrictive scopes.

Azure Cosmos DB credentials

You can use these credentials to authenticate the following nodes:

- [Azure Cosmos DB](#)

Prerequisites

- Create an [Azure](#) subscription.
- Create an [Azure Cosmos DB account](#).

Supported authentication methods

- API Key

Related resources

Refer to [Azure Cosmos DB's API documentation](#) for more information about the service.

Using API Key

To configure this credential, you'll need:

- An **Account**: The name of your Azure Cosmos DB account.
- A **Key**: A key for your Azure Cosmos DB account. Select **Overview** > **Keys** in the Azure portal for your Azure Cosmos DB. You can use either of the two account keys for this purpose.
- A **Database**: The name of the Azure Cosmos DB database to connect to.

Refer to [Get your primary key | Microsoft](#) for more detailed steps.

Common issues

Here are the known common errors and issues with Azure Cosmos DB credentials.

-8<- “_snippets/integrations/builtin/credentials/microsoft-need-admin-approval.md”

Azure AI Search credentials

You can use these credentials to authenticate the following nodes:

- [Azure AI Search Vector Store](#)

Prerequisites

- An [Azure subscription](#)
- An Azure AI Search service created in the [Azure Portal](#)

Supported authentication methods

This node uses API key authentication.

Related resources

Refer to [Azure AI Search documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- **Endpoint:** Your Azure AI Search service URL (format: `https://your-service.search.windows.net`)
- **API Key:** Admin key (read-write) or query key (read-only)

To get these values:

1. Navigate to your Azure AI Search service in the [Azure Portal](#)
2. Copy the **URL** from the **Overview** section
3. Go to **Settings > Keys** and copy:
 - **Admin key** for full read-write access, or
 - **Query key** for read-only querying
4. Enter these values in n8n

Troubleshooting

Authentication errors

API key authentication fails: - Verify the API key is correct and hasn't been regenerated in Azure Portal - Confirm you're using an admin key for write operations (insert/update) - Check that the key hasn't expired or been rotated

Connection issues

- Verify endpoint URL format: `https://your-service.search.windows.net`
- Confirm your Azure AI Search service is running
- Check network security rules and firewall settings allow access

from your n8n instance

Azure Storage credentials

You can use these credentials to authenticate the following nodes:

- [Azure Storage](#)

Prerequisites

- Create an [Azure subscription](#).
- Create an [Azure storage account](#).

Supported authentication methods

- OAuth2
- Shared Key

Related resources

Refer to [Azure Storage's API documentation](#) for more information about the service.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

For self-hosted users, there are two main steps to configure OAuth2 from scratch:

1. [Register an application](#) with the Microsoft Identity Platform.
2. [Generate a client secret](#) for that application.

Follow the detailed instructions for each step below. For more detail on the Microsoft OAuth2 web flow, refer to [Microsoft authentication and authorization basics](#).

Register an application

Register an application with the Microsoft Identity Platform:

1. Open the [Microsoft Application Registration Portal](#).
2. Select **Register an application**.
3. Enter a **Name** for your app.
4. In **Supported account types**, select **Accounts in any organizational directory (Any Azure AD directory - Multi-tenant) and personal Microsoft accounts (for example, Skype, Xbox)**.
5. In **Register an application**:
 1. Copy the **OAuth Callback URL** from your n8n credential.
 2. Paste it into the **Redirect URI (optional)** field.

3. Select **Select a platform > Web**.
6. Select **Register** to finish creating your application.
7. Copy the **Application (client) ID** and paste it into n8n as the **Client ID**.

Refer to [Register an application with the Microsoft Identity Platform](#) for more information.

Generate a client secret

With your application created, generate a client secret for it:

1. On your Microsoft application page, select **Certificates & secrets** in the left navigation.
2. In **Client secrets**, select **+ New client secret**.
3. Enter a **Description** for your client secret, such as n8n credential.
4. Select **Add**.
5. Copy the **Secret** in the **Value** column.
6. Paste it into n8n as the **Client Secret**.
7. Select **Connect my account** in n8n to finish setting up the connection.
8. Log in to your Microsoft account and allow the app to access your info.

Refer to Microsoft's [Add credentials](#) for more information on adding a client secret.

Using Shared Key

To configure this credential, you'll need:

- An **Account**: The name of your Azure Storage account.
- A **Key**: A shared key for your Azure Storage account. Select **Security + networking** and then **Access keys**. You can use either of the two account keys for this purpose.

Refer to [Manage storage account access keys | Microsoft](#) for more detailed steps.

Common issues

Here are the known common errors and issues with Azure Storage credentials.

-8<- “_snippets/integrations/builtin/credentials/microsoft-need-admin-approval.md”

BambooHR credentials

You can use these credentials to authenticate the following node:

- [BambooHR](#)

Prerequisites

Create a [BambooHR](#) account.

Supported authentication methods

- API key

Related resources

Refer to [BambooHR's API documentation](#) for more information about the service.

Using API Key

To configure this credential, you'll need:

- Your BambooHR **Subdomain**: the part between `https://` and `.bamboohr.com`
 - A BambooHR **API Key**: Refer to the [Authentication section of BambooHR's Getting Started API documentation](#) for instructions on generating an API key.
-

Bannerbear credentials

You can use these credentials to authenticate the following nodes:

- [Bannerbear](#)

Prerequisites

Create a [Bannerbear](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Bannerbear's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- A **Project API Key**: To generate an API key, first create a Bannerbear project. Go to **Settings > API Key** to view the API

key. Refer to the [Bannerbear API Authentication documentation](#) for more detailed steps.

Baserow credentials

You can use these credentials to authenticate the following node:

- [Baserow](#)

Prerequisites

Create a [Baserow](#) account on any hosted Baserow instance or a self-hosted instance.

Supported authentication methods

- Basic auth

Related resources

Refer to [Baserow's documentation](#) for more information about the service.

Refer to [Baserow's auto-generated API documentation](#) for more information about the API specifically.

Using basic auth

To configure this credential, you'll need:

- Your Baserow **Host**
- A **Username** and **Password** to log in with

Follow these steps:

1. Enter the **Host** for the Baserow instance:
 - For a Baserow-hosted instance: leave as `https://api.baserow.io`.
 - For a self-hosted instance: set to your self-hosted instance API URL.
2. Enter the **Username** for the user account n8n should use.
3. Enter the **Password** for that user account.

Refer to [Baserow's API Authentication documentation](#) for information on creating user accounts.

Beeminder credentials

You can use these credentials to authenticate the following node:

- [Beeminder](#)

Prerequisites

Create a [Beeminder account](#).

Supported authentication methods

- API user token

Related resources

Refer to [Beeminder's API documentation](#) for more information about the service.

Using API user token

To configure this credential, you'll need:

- A **User** name: Should match the user who the Auth Token is generated for.
- A personal **Auth Token** for that user. Generate this using either method below:
 - In the GUI: From the [Apps & API](#) option within **Account Settings**
 - In the API: From hitting the [auth_token API endpoint](#)

Bitbucket credentials

You can use these credentials to authenticate the following nodes:

- [Bitbucket Trigger](#)

Prerequisites

Create a [Bitbucket account](#).

Supported authentication methods

- Access token

Related resources

Refer to [Bitbucket's API documentation](#) for more information about the service.

Configuring Bitbucket access token

1. Log in to Bitbucket and open your account or personal settings.

2. Find the section for API tokens or security settings.
3. Create a new API token, giving it a name and expiry date that matches your use case.
4. Select Bitbucket as the app, then choose the required scopes (permissions):

```
read:user:bitbucket  
read:workspace:bitbucket  
read:repository:bitbucket  
read:webhook:bitbucket  
write:webhook:bitbucket  
delete:webhook:bitbucket
```

5. Review and create the token. Copy the generated token and add it to n8n. Bitbucket only shows the token once.

For detailed instructions, see [Create an API token](#).

Bitly credentials

You can use these credentials to authenticate the following node:

- [Bitly](#)

Prerequisites

Create a [Bitly](#) account.

Supported authentication methods

- API token
- OAuth2

Related resources

Refer to [Bitly's API documentation](#) for more information about the service.

Using API token

To configure this credential, you'll need:

- An **Access Token**: Once logged in, visit [Settings > Developer Settings > API](#) to generate an Access Token.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you need to configure OAuth2 from scratch or need more detail on what's happening in the OAuth web flow, refer to the [Bitly API Authentication documentation](#) for more information.

Bitwarden credentials

You can use these credentials to authenticate the following node:

- [Bitwarden](#)

Prerequisites

Create a [Bitwarden](#) Teams organization or Enterprise organization account. (Bitwarden only makes the Bitwarden Public API available for these [organization](#) plans.)

Supported authentication methods

- API key

Related resources

Refer to [Bitwarden's Public API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- A **Client ID**: Provided when you generate an API key
- A **Client Secret**: Provided when you generate an API key
- The **Environment**:
 - Choose **Cloud-hosted** if you don't self-host Bitwarden. No further configuration required.
 - Choose **Self-hosted** if you host Bitwarden on your own server. Enter your **Self-hosted domain** in the appropriate field.

The Client ID and Client Secret must be for an **Organization API Key**, not a Personal API Key. Refer to the [Bitwarden Public API Authentication documentation](#) for instructions on generating an Organization API Key.

Box credentials

You can use these credentials to authenticate the following nodes:

- [Box](#)
- [Box Trigger](#)

Prerequisites

Create a [Box](#) account.

Supported authentication methods

- OAuth2

Related resources

Refer to [Box's API documentation](#) for more information about the service.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you need to configure OAuth2 from scratch or need more detail on what's happening in the OAuth web flow, you'll need to create a Custom App. Refer to the [Box OAuth2 Setup documentation](#) for more information.

Brandfetch credentials

You can use these credentials to authenticate the following node:

- [Brandfetch](#)

Prerequisites

Create a [Brandfetch developer](#) developer account.

Supported authentication methods

- API key

Related resources

Refer to [Brandfetch's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Refer to the [Brandfetch Create an Account documentation](#) to generate an API key.

Brevo credentials

You can use these credentials to authenticate the following nodes:

- [Brevo node](#)
- [Brevo Trigger node](#)

Prerequisites

Create a [Brevo](#) developer account.

Supported authentication methods

- API key

Related resources

Refer to [Brevo's API documentation](#) for more information about authenticating with the service.

API key

To configure this credential, you'll need:

- An **API Key**: Refer to the [Brevo API Quick Start documentation](#) for instructions on creating a new API key.
-

Bubble credentials

You can use these credentials to authenticate the following nodes:

- [Bubble](#)

Supported authentication methods

- API key

Related resources

Refer to [Bubble's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need a paid [Bubble](#) account and:

- An **API Token**
- An **App Name**
- Your **Domain**, if you're using a custom domain

To set it up, you'll need to create an app:

1. Go to the [Apps](#) page in Bubble.
2. Select **Create an app**.
3. Enter a **Name** for your app, like n8n-integration.
4. Select **Get started**. The app's details open.
5. In the left navigation, select **Settings** (the gear cog icon).
6. Select the **API** tab.
7. In the **Public API Endpoints** section, check the box to **Enable Data API**.
8. The page displays the **Data API root URL**, for example: <https://n8n-integration.bubbleapps.io/version-test/api/1.1/obj>.
9. Copy the part of the URL after `https://` and before `.bubbleapps.io` and enter it in n8n as the **App Name**. In the above example, you'd enter `n8n-integration`.
10. Select **Generate a new API token**.
11. Enter an **API Token Label**, like `n8n integration`.
12. Copy the **Private key** and enter it as the **API Token** in your n8n credential.
 - Refer to [Data API | Authentication](#) for more information on generating API tokens.
13. In n8n, select the **Environment** that best matches your app:
 - Select **Development** for an app that you haven't deployed, accessed at <https://appname.bubbleapps.io/version-test> or <https://www.mydomain.com/version-test>.
 - Select **Live** for an app that you've [deployed](#), accessed at <https://appname.bubbleapps.io> or <https://www.mydomain.com>.
14. In n8n, select your **Hosting**:
 - If you haven't set up a custom domain, select **Bubble Hosting**.
 - If you've set up a [custom domain](#), select **Self Hosted** and enter your custom **Domain**.

Refer to Bubble's [Creating and managing apps](#) documentation for more information.

Cal.com credentials

You can use these credentials to authenticate the following nodes:

- [Cal.com Trigger](#)

Prerequisites

Create a [Cal.com](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Cal.com's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Refer to the [Cal API Quick Start documentation](#) for information on how to generate a new API key.
 - A **Host**: If you're using the cloud version of Cal.com, leave the Host as `https://api.cal.com`. If you're self-hosting Cal.com, enter the **Host** for your Cal.com instance.
-

Calendly credentials

You can use these credentials to authenticate the following nodes:

- [Calendly Trigger](#)

Supported authentication methods

- API access token
- OAuth2

Related resources

Refer to [Calendly's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need a [Calendly](#) account and:

- An API Key or **Personal Access Token**

To get your access token:

1. Go to the Calendly [Integrations & apps](#) page.
2. Select [API & Webhooks](#).
3. In **Your Personal Access Tokens**, select **Generate new token**.
4. Enter a **Name** for your access token, like `n8n integration`.
5. Select **Create token**.
6. Select **Copy token** and enter it in your n8n credential.

Refer to [Calendly's API authentication documentation](#) for more information.

Using OAuth2

To configure this credential, you'll need a [Calendly developer account](#) and:

- A **Client ID**
- A **Client Secret**

To get both, create a new OAuth app in Calendly:

1. Log in to Calendly's developer portal and go to [My apps](#).
2. Select **Create new app**.
3. Enter a **Name of app**, like n8n integration.
4. In **Kind of app**, select **Web**.
5. In **Environment type**, select the environment that corresponds to your usage, either **Sandbox** or **Production**.
 - Calendly recommends starting with **Sandbox** for development and creating a second application for **Production** when you're ready to go live.
6. Copy the **OAuth Redirect URL** from n8n and enter it as a **Redirect URI** in the OAuth app.
7. Select **Save & Continue**. The app details display.
8. Copy the **Client ID** and enter this as your n8n **Client ID**.
9. Copy the **Client secret** and enter this as your n8n **Client Secret**.
10. Select **Connect my account** in n8n and follow the on-screen prompts to finish authorizing the credential.

Refer to [Registering your application with Calendly](#) for more information.

Carbon Black credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

- Create a [Carbon Black subscription](#).
- Create a [Carbon Black developer account](#).

Authentication methods

- API key

Related resources

Refer to [Carbon Black's documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n's website.

Using API key

To configure this credential, you'll need:

- A **URL**: This URL is determined by the environment/product URL you use. You can find it by looking at the web address of your Carbon Black Cloud console. Refer to [Carbon Black's URL Parts documentation](#) for more information.
 - An **Access Token**: Refer to the [Carbon Black Create an API key documentation](#) to create an API key. Add the **API Secret Key** as the **Access Token** in n8n.
-

Chargebee credentials

You can use these credentials to authenticate the following nodes:

- [Chargebee](#)
- [Chargebee Trigger](#)

Prerequisites

Create a [Chargebee](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Chargebee's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **Account Name**: This is your Chargebee Site Name or subdomain, for example if `https://n8n.chargebee.com` is the full site name, the Account Name is `n8n`.
- An **API Key**: Refer to the [Chargebee Creating an API key documentation](#) for steps on how to generate an API key.

Refer to their more general [API authentication documentation](#) for further clarification.

CircleCI credentials

You can use these credentials to authenticate the following nodes:

- [CircleCI](#)

Prerequisites

Create a [CircleCI](#) account.

Supported authentication methods

- Personal API token

Related resources

Refer to [CircleCI's API documentation](#) for more information about the service.

Using personal API token

To configure this credential, you'll need:

- A **Personal API Token**: Refer to the [CircleCI Creating a Personal API token documentation](#) for instructions on creating your token.
-

Cisco Meraki credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

- Create a [Cisco DevNet developer account](#).
- Access to a [Cisco Meraki account](#).

Authentication methods

- API key

Related resources

Refer to [Cisco Meraki's API documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n's website.

Using API key

To configure this credential, you'll need:

- An **API Key**: Refer to the [Cisco Meraki Obtaining your Meraki API Key documentation](#) for instructions on getting your API Key.

Cisco Secure Endpoint credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

- Create a [Cisco DevNet developer account](#).
- Access to a [Cisco Secure Endpoint license](#).

Authentication methods

- OAuth2

Related resources

Refer to [Cisco Secure Endpoint’s documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n’s website.

Using OAuth2

To configure this credential, you’ll need:

- The **Region** for your Cisco Secure Endpoint. Options are:
 - Asia Pacific, Japan, and China
 - Europe
 - North America
- A **Client ID**: Provided when you register a SecureX API Client
- A **Client Secret**: Provided when you register a SecureX API Client

To get a Client ID and Client Secret, you’ll need to Register a SecureX API Client. Refer to [Cisco Secure Endpoint’s authentication documentation](#) for detailed instructions. Use the SecureX **Client Password** as the **Client Secret** within the n8n credential.

Cisco Umbrella credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

- Create a [Cisco DevNet developer account](#).
- A [Cisco Umbrella user account](#) with **Full Admin** role.

Authentication methods

- API key

Related resources

Refer to [Cisco Umbrella's API documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n's website.

Using API key

To configure this credential, you'll need:

- An **API Key**
- A **Secret**: Provided when you generate an API key

Refer to the [Cisco Umbrella Manage API Keys documentation](#) for instructions on creating an Umbrella API key.

Clearbit credentials

You can use these credentials to authenticate the following node:

- [Clearbit](#)

Prerequisites

Create a [Clearbit](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Clearbit's API documentation](#) for more information about authenticating with the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Refer to [Clearbit's API Authentication documentation](#) for more information on creating and viewing API keys.
-

ClickUp credentials

You can use these credentials to authenticate the following nodes:

- [ClickUp](#)
- [ClickUp Trigger](#)

Supported authentication methods

- API access token
- OAuth2

Related resources

Refer to [ClickUp's documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need a [ClickUp](#) account and:

- A Personal API Access Token

To get your personal API token:

1. If you're using ClickUp 2.0, select your avatar in the lower-left corner and select **Apps**. If you're using ClickUp 3.0, select your avatar in the upper-right corner, select **Settings**, and scroll down to select **Apps** in the sidebar.
2. Under **API Token**, select **Generate**.
3. Copy your **Personal API token** and enter it in your n8n credential as the **Access Token**.

Refer to [ClickUp's Personal Token documentation](#) for more information.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you're [self-hosting](#) n8n, you'll need to create an OAuth app:

1. In ClickUp, select your avatar and select **Integrations**.
2. Select **ClickUp API**.
3. Select **Create an App**.
4. Enter a **Name** for your app.
5. In n8n, copy the **OAuth Redirect URL**. Enter this as your ClickUp app's **Redirect URL**.
6. Once you create your app, copy the **client_id** and **secret** and enter them in your n8n credential.
7. Select **Connect my account** and follow the on-screen prompts to finish connecting the credential.

Refer to the [ClickUp Oauth flow documentation](#) for more information.

Clockify credentials

You can use these credentials to authenticate the following nodes:

- [Clockify](#)
- [Clockify Trigger](#)

Prerequisites

Create a [Clockify](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Clockify's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Access your API key from your [Clockify Profile Settings](#).

Cloudflare credentials

You can use these credentials to authenticate the following nodes:

- [Cloudflare node](#)

Prerequisites

- Create a [Cloudflare account](#).
- [Add a domain](#).

Supported authentication methods

- API token

Related resources

Refer to [Cloudflare's API documentation](#) for more information about the service.

Using API token

To configure this credential, you'll need:

- An **API token**: Follow the [Cloudflare documentation to create an API token](#).
-

Cockpit credentials

You can use these credentials to authenticate the following nodes:

- [Cockpit](#)

Prerequisites

- Create a [Cockpit](#) account.
- Set up a [self-hosted instance of Cockpit](#).

Supported authentication methods

- API access token

Related resources

Refer to [Cockpit's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need:

- Your **Cockpit URL**: The URL you use to access your Cockpit instance
 - An **Access Token**: Refer to the [Cockpit Managing tokens documentation](#) for instructions on creating an API token. Use the **API token** as the n8n **Access Token**.
-

Coda credentials

You can use these credentials to authenticate the following nodes:

- [Coda](#)

Prerequisites

Create a [Coda](#) account.

Supported authentication methods

- API access token

Related resources

Refer to [Coda's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need:

- An **API Access Token**: Generate an API access token in your Coda [Account settings](#).
-

Cohere credentials

You can use these credentials to authenticate the following nodes:

- [Cohere](#)
- [Cohere Chat](#)
- [Reranker Cohere](#)
- [Embeddings Cohere](#)

Prerequisites

Create a [Cohere](#) account.

You'll need an account with the following access:

- For the Trial API, you need User or Owner permissions.
- For Production API, you need Owner permissions.

Refer to [Cohere Teams and Roles documentation](#) for more information.

Supported authentication methods

- API key

Related resources

Refer to [Cohere's documentation](#) for more information about the service.

-8<- “ snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Using API key

To configure this credential, you'll need:

- An **API Key**: To generate a Cohere API key, go to the [API Keys](#) section of your Cohere dashboard.
-

Contentful credentials

You can use these credentials to authenticate the following nodes:

- [Contentful](#)

Prerequisites

- Create a [Contentful](#) account.
- Create a [Contentful space](#).

Supported authentication methods

- API access token

Related resources

Refer to [Contentful's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need:

- Your Contentful **Space ID**: The Space ID displays as you generate the tokens; You can also refer to the [Contentful Find space ID documentation](#) to view the Space ID.
- A **Content Delivery API Access Token**: Required if you want to use the [Content Delivery API](#). Leave blank if you don't intend to use this API.
- A **Content Preview API Access Token**: Required if you want to use the [Content Preview API](#). Leave blank if you don't intend to use this API.

View and generate access tokens in Contentful in **Settings > API keys**. Contentful generates tokens for both Content Delivery API and Content Preview API as part of a single key. Refer to the [Contentful API authentication documentation](#) for detailed instructions.

Convert API credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Supported authentication methods

- API Token

Related resources

Refer to [ConvertAPI’s API documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n’s website.

Using API Token

To configure this credential, you’ll need a [ConvertAPI](#) account and:

- An [**API Token**](#) to authenticate requests to the service.

Refer to [ConvertAPI’s API documentation](#) for more information about authenticating to the service.

ConvertKit credentials

You can use these credentials to authenticate the following nodes:

- [ConvertKit](#)
- [ConvertKit Trigger](#)

Prerequisites

Create a [ConvertKit](#) account.

Supported authentication methods

- API key

Related resources

Refer to [ConvertKit’s API documentation](#) for more information about the service.

Using API key

To configure this credential, you’ll need:

- An **API Secret**: Access your ConvertKit API key in [Account](#)

[Settings > Advanced](#). Add this key as the **API Secret** in n8n.

Copper credentials

You can use these credentials to authenticate the following nodes:

- [Copper](#)
- [Copper Trigger](#)

Prerequisites

Create a [Copper](#) account at the **Professional** or **Business** plan level.

Supported authentication methods

- API key

Related resources

Refer to [Copper's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Refer to the [Copper Generating an API key documentation](#) for information on generating an API key.
 - An **Email** address: Use the API key creator's email address
-

Cortex credentials

You can use these credentials to authenticate the following nodes:

- [Cortex](#)

Prerequisites

Install [Cortex](#) on your server.

Supported authentication methods

- API key

Related resources

Refer to [Cortex's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Refer to the [Cortex API Authentication documentation](#) for detailed instructions on generating API keys.
 - The URL/Server Address for your **Cortex Instance** (defaults to `http://<your_server_address>:9001/`)
-

CratedB credentials

You can use these credentials to authenticate the following nodes:

- [CratedB](#)

Prerequisites

An available instance of CrateDB.

Supported authentication methods

- account connection

Related resources

Refer to [CratedB's documentation](#) for more information about the service.

Using account connection

To configure this credential, you'll need:

- Your **Host** name
- Your **Database** name
- A **User** name
- A user **Password**
- To set the **SSL** parameter. Refer to the [CratedB Secured Communications \(SSL/TLS\) documentation](#) for more information.
The options n8n supports are:
 - Allow
 - Disable
 - Require
- A **Port** number

Refer to the [Connect to a CrateDB cluster documentation](#) for detailed instructions on these fields and their default values.

crowd.dev credentials

You can use these credentials to authenticate the following nodes:

- [crowd.dev](#)
- [crowd.dev Trigger](#)

Prerequisites

Create a working instance of [crowd.dev](#).

Supported authentication methods

- API key

Related resources

Refer to [crowd.dev's documentation](#) for more information about the service, and their [API documentation](#) for working with the API.

Using API key

To configure this credential, you'll need:

- A **URL**:
 - If your crowd.dev instance is hosted on crowd.dev, keep the default of `https://app.crowd.dev`.
 - If your crowd.dev instance is self-hosted, use the URL you use to access your crowd.dev instance.
- Your crowd.dev **Tenant ID**: Displayed in the **Settings** section of the crowd.dev app
- An API **Token**: Displayed in the **Settings** section of the crowd.dev app

Refer to the [crowd.dev API documentation](#) for more detailed instructions.

CrowdStrike credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create a [CrowdStrike](#) account.

Authentication methods

- OAuth2

Related resources

Refer to CrowdStrike's documentation for more information about the service. Their documentation is behind a log in, so you must log in to your account on their website to access the API documentation.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n's website.

Using OAuth2

To configure this credential, you'll need:

- The **URL** of your CrowdStrike instance
- A **Client ID**: Generated by creating a new API Client in CrowdStrike in **Support > API Clients and Keys**.
- A **Client Secret**: Generated by creating a new API Client in CrowdStrike in **Support > API Clients and Keys**.

When setting up your API client, grant it the `usermgmt:read` scope. n8n relies on this to test that the credential is working.

A broad outline of the appropriate steps is available publicly at the CrowdStrike blog: [Getting Access to the CrowdStrike API](#). CrowdStrike's full documentation is behind a log in, so you must log in to your account to access the full API documentation.

Customer.io credentials

You can use these credentials to authenticate the following nodes with Customer.io.

- [Customer.io](#)
- [Customer.io Trigger](#)

Prerequisites

Create a [Customer.io](#) account.

Supported authentication methods

- API Key

Related resources

Refer to [Customer.io's summary API documentation](#) for more information about the service.

For detailed API reference documentation for each API, refer to the [Track API documentation](#) and the [App API documentation](#).

Using API key

To configure this credential, you'll need:

- A **Tracking API Key**: For use with the [Track API](https://track.customer.io/api/v1/) at <https://track.customer.io/api/v1/>. See the FAQs below for more details.
- Your **Region**: Customer.io uses different API subdomains depending on the region you select. Options include:
 - **Global region**: Keeps the default URLs for both APIs; for use in all non-EU countries/regions.
 - **EU region**: Adjusts the Track API subdomain to `track-eu` and the App API subdomain to `api-eu`; only use this if you are in the EU.
- A **Tracking Site ID**: Required with your **Tracking API Key**
- An **App API Key**: For use with the [App API](https://api.customer.io/v1/api/) at <https://api.customer.io/v1/api/>. See the FAQs below for more details.

Refer to the [Customer.io Finding and managing your API credentials documentation](#) for instructions on creating both Tracking API and App API keys.

Why you need a Tracking API Key and an App API Key

Customer.io has two different API endpoints and generates and stores the keys for each slightly differently:

- The [Track API](https://track.customer.io/api/v1/) at <https://track.customer.io/api/v1/>
- The [App API](https://api.customer.io/v1/api/) at <https://api.customer.io/v1/api/>

The Track API requires a Tracking Site ID; the App API doesn't.

Based on the operation you want to perform, n8n uses the correct API key and its corresponding endpoint.

Datadog credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create a [Datadog account](#).

Related resources

Refer to [Datadog's API documentation](#) for more information about authenticating with the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n's website.

Using API Key

To configure this credential, you'll need:

- Your Datadog instance **Host**
- An **API Key**
- An **App Key**

Refer to [Authentication](#) on Datadog's website for more information.

DeepL credentials

You can use these credentials to authenticate the following nodes:

- [DeepL](#)

Prerequisites

Create a [DeepL developer](#) account. n8n works with both Free and Pro API Plans.

Supported authentication methods

- API key

Related resources

Refer to [DeepL's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Refer to [DeepL's Authentication documentation](#) for more information on getting your API key.
- To identify which **API Plan** you're on. DeepL has different API endpoints for each plan, so be sure you select the correct one:
 - Pro Plan
 - Free Plan

DeepSeek credentials

You can use these credentials to authenticate the following nodes:

- [Chat DeepSeek](#)

Prerequisites

Create a [DeepSeek](#) account.

Supported authentication methods

- API key

Related resources

Refer to [DeepSeek's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**

To generate your API Key:

1. Login to your DeepSeek account or [create](#) an account.
2. Open your [API keys](#) page.
3. Select **Create new secret key** to create an API key, optionally naming the key.
4. Copy your key and add it as the **API Key** in n8n.

Refer to the [Your First API Call](#) page for more information.

Demio credentials

You can use these credentials to authenticate the following nodes:

- [Demio](#)

Prerequisites

Create a [Demio](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Demio's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**
- An **API Secret**

You must have Owner status in Demio to generate API keys and secrets. To view and generate API keys and secrets, go to **Account Settings > API**. Refer to the [Demio Account Owner Settings documentation](#) for more detailed steps.

DFIR-IRIS credentials

-8<- “snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

An accessible instance of [DFIR-IRIS](#).

Related resources

Refer to [DFIR-IRIS’s API documentation](#) for more information about authenticating with the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows](#) and related content on n8n’s website.

Using API Key

To configure this credential, you’ll need:

- An **API Key**: Refer to [DFIR-IRIS’s API documentation](#) for instructions on getting your API key.
 - The **Base URL** of your DFIR-IRIS instance.
-

DHL credentials

You can use these credentials to authenticate the following nodes:

- [DHL](#)

Supported authentication methods

- API key

Related resources

Refer to [DHL’s Developer documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need a [DHL Developer](#) account and:

- An **API Key**

To get an API key, create an app:

1. In the DHL Developer portal, select the user icon to open your [User Apps](#).
2. Select **+ Create App**.
3. Enter an **App name**, like n8n integration.
4. Enter a **Machine name**, like n8n_integration.
5. In **SELECT APIs**, select **Shipment Tracking - Unified**. The API is added to the **Add API to app** section.
6. In the **Add API to app** section, select the **+** next to the **Shipment Tracking - Unified** API.
7. Select **Create App**. The **Apps** page opens, displaying the app you just created.
8. Select the app you just created to view its details.
9. Select **Show key** next to **API Key**.
10. Copy the **API Key** and enter it in your n8n credential.

Refer to [How to create an app?](#) for more information.

Discord credentials

You can use these credentials to authenticate the following nodes:

- [Discord](#)

Prerequisites

- Create a [Discord](#) account.
- For Bot and OAuth2 credentials:
 - [Set up your local developer environment](#).
 - [Create an application and a bot user](#).
- For webhook credentials, [create a webhook](#).

Supported authentication methods

- Bot
- OAuth2
- Webhook

Not sure which method to use? Refer to [Choose an authentication method](#) for more guidance.

Related resources

Refer to [Discord's Developer documentation](#) for more information about the service.

Using bot

Use this method if you want to add the bot to your Discord server using a bot token rather than OAuth2.

To configure this credential, you'll need:

- A **Bot Token**: Generated once you create an application with a bot.

To create an application with a bot and generate the **Bot Token**:

1. If you don't have one already, create an app in the [developer portal](#).
2. Enter a **Name** for your app.
3. Select **Create**.
4. Select **Bot** from the left menu.
5. Under **Token**, select **Reset Token** to generate a new bot token.
6. Copy the token and add it to your n8n credential.
7. In **Bot > Privileged Gateway Intents**, add any privileged intents you want your bot to have. Refer to [Configuring your bot](#) for more information on privileged intents.
 - n8n recommends activating **SERVER MEMBERS INTENT: Required for your bot to receive events listed under GUILD_MEMBERS**.
8. In **Installation > Installation Contexts**, select the installation contexts you want your bot to use:
 - Select **Guild Install** for server-installed apps. (Most common for n8n users.)
 - Select **User Install** for user-installed apps. (Less common for n8n users, but may be useful for testing.)
 - Refer to Discord's [Choosing installation contexts](#) documentation for more information about these installation contexts.
9. In **Installation > Install Link**, select **Discord Provided Link** if it's not already selected.
10. Still on the **Installation** page, in the **Default Install Settings** section, select applications.commands and bot scopes. Refer to Discord's [Scopes](#) documentation for more information about these and other scopes.
11. Add permissions on the **Bot > Bot Permissions** page. Refer to Discord's [Permissions](#) documentation for more information. n8n recommends selecting these permissions for the **Discord** node:
 - Manage Roles
 - Manage Channels
 - Read Messages/View Channels
 - Send Messages
 - Create Public Threads
 - Create Private Threads
 - Send Messages in Threads
 - Send TTS Messages
 - Manage Messages
 - Manage Threads
 - Embed Links
 - Attach Files
 - Read Message History
 - Add Reactions
12. Add the app to your server or test server:
 1. Go to **Installation > Install Link** and copy the link listed there.

2. Paste the link in your browser and hit Enter.
3. Select **Add to server** in the installation prompt.
4. Once your app's added to your server, you'll see it in the member list.

These steps outline the basic functionality needed to set up your n8n credential. Refer to the [Discord Creating an App](#) guide for more information on creating an app, especially:

- [Fetching your credentials](#) for getting your app's credentials into your local developer environment.
- [Handling interactivity](#) for information on setting up public endpoints for interactive /slash commands.

Using OAuth2

Use this method if you want to add the bot to Discord servers using the OAuth2 flow, which simplifies the process for those installing your app.

To configure this credential, you'll need:

- A **Client ID**
- A **Client Secret**
- Choose whether to send **Authentication** in the **Header** or **Body**
- A **Bot Token**

For details on creating an application with a bot and generating the token, follow the same steps as in [Using bot](#) above.

Then:

1. Copy the **Bot Token** you generate and add it into the n8n credential.
2. Open the **OAuth2** page in your Discord application to access your **Client ID** and generate a **Client Secret**. Add these to your n8n credential.
3. From n8n, copy the **OAuth Redirect URL** and add it into the Discord application in **OAuth2 > Redirects**. Be sure you save these changes.

Using webhook

To configure this credential, you'll need:

- A **Webhook URL**: Generated once you create a webhook.

To get a Webhook URL, you create a webhook and copy the URL that gets generated:

1. Open your Discord **Server Settings** and open the **Integrations** tab.
2. Select **Create Webhook** to create a new webhook.
3. Give your webhook a **Name** that makes sense.
4. Select the **avatar** next to the **Name** to edit or upload a new avatar.
5. In the **CHANNEL** dropdown, select the channel the webhook should post to.
6. Select **Copy Webhook URL** to copy the Webhook URL. Enter this

URL in your n8n credential.

Refer to the [Discord Making a Webhook documentation](#) for more information.

Choose an authentication method

The simplest installation is a **webhook**. You create and add webhooks to a single channel on a Discord server. Webhooks can post messages to a channel. They don't require a bot user or authentication. But they can't listen or respond to user requests or commands. If you need a straightforward way to send messages to a channel without the need for interaction or feedback, use a webhook.

A **bot** is an interactive step up from a webhook. You add bots to the Discord server (referred to as a **guild** in the Discord API documentation) or to user accounts. Bots added to the server can interact with users on all the server's channels. They can manage channels, send and retrieve messages, retrieve the list of all users, and change their roles. If you need to build an interactive, complex, or multi-step workflow, use a bot.

OAuth2 is basically a **bot** that uses an OAuth2 flow rather than just the bot token. As with bots, you add these to the Discord server or to user accounts. These credentials offer the same functionalities as bots, but they can simplify the installation of the bot on your server.

Discourse credentials

You can use these credentials to authenticate the following nodes:

- [Discourse](#)

Prerequisites

- Host an instance of [Discourse](#)
- Create an account on your hosted instance and make sure that you are an admin

Supported authentication methods

- API key

Related resources

Refer to [Discourse's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- The **URL** of your Discourse instance, for example <https://community.n8n.io>
- An **API Key**: Create an API key through the Discourse admin panel. Refer to the [Discourse create and configure an API key documentation](#) for instructions on creating an API key and specifying a username.
- A **Username**: Use your own name, system, or another user.

Refer to the Authentication section of the [Discourse API documentation](#) for examples.

Disqus credentials

You can use these credentials to authenticate the following nodes:

- [Disqus](#)

Prerequisites

- Create a [Disqus](#) account.
- Register an [API application](#).

Supported authentication methods

- API access token

Related resources

Refer to [Disqus's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need:

- An **Access Token**: Once you've registered an API application, copy the **API Key** and add it to n8n as the **Access Token**.
-

Drift credentials

You can use these credentials to authenticate the following nodes:

- [Drift](#)

Prerequisites

- Create a [Drift](#) account.
- [Create a Drift app](#).

Supported authentication methods

- API personal access token
- OAuth2

Related resources

Refer to [Drift's API documentation](#) for more information about the service.

Using API personal access token

To configure this credential, you'll need:

- A **Personal Access Token**: To get a token, [create a Drift app](#). [Install the app](#) to generate an OAuth Access token. Add this to the n8n credential as your **Personal Access Token**.

Using OAuth2

-8<- “snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you need to configure OAuth2 from scratch or need more detail on what's happening in the OAuth web flow, refer to the instructions in the [Drift Authentication and Scopes documentation](#) to set up OAuth for your app.

Dropbox credentials

You can use these credentials to authenticate the following nodes:

- [Dropbox](#)

Supported authentication methods

- API access token: Dropbox recommends this method for testing with your user account and granting a limited number of users access.
- OAuth2: Dropbox recommends this method for production or for testing with more than 50 users.

Related resources

Refer to [Dropbox's Developer documentation](#) for more information about the service.

Using access token

To configure this credential, you'll need a [Dropbox](#) developer account and:

- An **Access Token**: Generated once you create a Dropbox app.
- An **App Access Type**

To set up the credential, create a Dropbox app:

1. Open the [App Console](#) within the Dropbox developer portal.
2. Select **Create app**.
3. In **Choose an API**, select **Scoped access**.
4. In **Choose the type of access you need**, choose whichever option best fits your use of the [Dropbox](#) node:
 - **App Folder** grants access to a single folder created specifically for your app.
 - **Full Dropbox** grants access to all files and folders in your user's Dropbox.
 - Refer to the [DBX Platform developer guide](#) for more information.
5. In **Name your app**, enter a name for your app, like n8n integration.
6. Check the box to agree to the **Dropbox API Terms and Conditions**.
7. Select **Create app**. The app's **Settings** open.
8. In the **OAuth 2** section, in **Generated access token**, select **Generate**.
9. Copy the access token and enter it as the **Access Token** in your n8n credential.
10. In n8n, select the same **App Access Type** you selected for your app.

Refer to the [Dropbox App Console Settings documentation](#) for more information.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

Cloud users need to select the **App Access Type**:

- **App Folder** grants access to a single folder created specifically for your app.
- **Full Dropbox** grants access to all files and folders in your user's Dropbox.
- Refer to the [DBX Platform developer guide](#) for more information.

If you're [self-hosting](#) n8n, you'll need to configure OAuth2 manually:

1. Open the [App Console](#) within the Dropbox developer portal.
2. Select **Create app**.
3. In **Choose an API**, select **Scoped access**.
4. In **Choose the type of access you need**, choose whichever option best fits your use of the [Dropbox](#) node:
 - **App Folder** grants access to a single folder created specifically for your app.
 - **Full Dropbox** grants access to all files and folders in your user's Dropbox.
 - Refer to the [DBX Platform developer guide](#) for more

information.

5. In **Name your app**, enter a name for your app, like n8n integration.
6. Check the box to agree to the **Dropbox API Terms and Conditions**.
7. Select **Create app**. The app's **Settings** open.
8. Copy the **App key** and enter it as the **Client ID** in your n8n credential.
9. Copy the **Secret** and enter it as the **Client Secret** in your n8n credential.
10. In n8n, copy the **OAuth Redirect URL** and enter it in the **Dropbox Redirect URIs**.
11. In n8n, select the same **App Access Type** you selected for your app.

Refer to the instructions in the [Dropbox Implementing OAuth documentation](#) for more information.

For internal tools and limited usage, you can keep your app private. But if you'd like your app to be used by more than 50 users or you want to distribute it, you'll need to complete Dropbox's production approval process. Refer to **Production Approval** in the [DBX Platform developer guide](#) for more information.

Dropcontact credentials

You can use these credentials to authenticate the following nodes:

- [Dropcontact](#)

Prerequisites

Create a developer account in [Dropcontact](#).

Supported authentication methods

- API key

Related resources

Refer to [Dropcontact's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: To view your API key in Dropcontact, go to [API](#). Refer to the [Dropcontact API key documentation](#) for more information.
-

Dynatrace credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create a [Dynatrace](#) account.

Related resources

Refer to [Dynatrace’s API documentation](#) for more information about authenticating with the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n’s website.

Using Access Token

To configure this credential, you’ll need:

- An **Access Token**

Refer to [Access Tokens](#) on Dynatrace’s website for more information.

E-goi credentials

You can use these credentials to authenticate the following nodes:

- [E-goi](#)

Prerequisites

Create an [E-goi](#) account.

Supported authentication methods

- API key

Related resources

Refer to [E-goi’s API documentation](#) for more information about the service.

Using API key

To configure this credential, you’ll need:

- An **API Key**: Refer to [E-goi's API key documentation](#) for instructions on generating and viewing an API key.
-

Elasticsearch credentials

You can use these credentials to authenticate the following nodes:

- [Elasticsearch](#)

Supported authentication methods

- Basic auth

Related resources

Refer to [Elasticsearch's documentation](#) for more information about the service.

Using basic auth

To configure this credential, you'll need an [Elasticsearch](#) account with a [deployment](#) and:

- A **Username**
- A **Password**
- Your Elasticsearch application's **Base URL** (also known as the Elasticsearch application endpoint)

To set up the credential:

1. Enter your Elasticsearch **Username**.
 2. Enter your Elasticsearch **Password**.
 3. In Elasticsearch, go to **Deployments**.
 4. Select your deployment.
 5. Select **Manage this deployment**.
 6. In the **Applications** section, copy the endpoint of the **Elasticsearch** application.
 7. Enter this in n8n as the **Base URL**.
 8. By default, n8n connects only if SSL certificate validation succeeds. If you'd like to connect even if SSL certificate validation fails, turn on **Ignore SSL Issues**.
-

Elastic Security credentials

You can use these credentials to authenticate the following nodes:

- [Elastic Security](#)

Prerequisites

- Create an [Elastic Security](#) account.
- [Deploy](#) an application.

Supported authentication methods

- Basic auth
- API Key

Related resources

Refer to [Elastic Security's documentation](#) for more information about the service.

Using basic auth

To configure this credential, you'll need:

- A **Username**: For the user account you log into Elasticsearch with.
- A **Password**: For the user account you log into Elasticsearch with.
- Your Elasticsearch application's **Base URL** (also known as the Elasticsearch application endpoint):
 1. In Elasticsearch, select the option to **Manage this deployment**.
 2. In the **Applications** section, copy the endpoint of the **Elasticsearch** application.
 3. Add this in n8n as the **Base URL**.

Using API key

To configure this credential, you'll need:

- An **API Key**: For the user account you log into Elasticsearch with. Refer to Elasticsearch's [Create API key documentation](#) for more information.
- Your Elasticsearch application's **Base URL** (also known as the Elasticsearch application endpoint):
 1. In Elasticsearch, select the option to **Manage this deployment**.
 2. In the **Applications** section, copy the endpoint of the **Elasticsearch** application.
 3. Add this in n8n as the **Base URL**.

Emelia credentials

You can use these credentials to authenticate the following nodes:

- [Emelia](#)
- [Emelia Trigger](#)

Prerequisites

Create an [Emelia](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Emelia's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: To generate an API Key in Emelia, access your **API Keys** by selecting the avatar in the top right (your **Settings**). Refer to the Authentication section of [Emelia's API documentation](#) for more information.
-

ERPNext credentials

You can use these credentials to authenticate the following nodes:

- [ERPNext](#)

Prerequisites

- Create an [ERPNext](#) account.

Supported authentication methods

- API key

Related resources

Refer to [ERPNext's documentation](#) for more information about the service.

Refer to [ERPNext's developer documentation](#) for more information about working with the framework.

Using API key

To configure this credential, you'll need:

- An **API Key**: Generate this from your own ERPNext user account in **Settings > My Settings > API Access**.
- An **API Secret**: Generated with the API key.
- Your ERPNext **Environment**:
 - For **Cloud-hosted**:
 - Your ERPNext **Subdomain**: Refer to the [FAQs](#)
 - Your **Domain**: Choose between `erpnext.com` and `frappe.cloud`.
 - For **Self-hosted**:
 - The fully qualified **Domain** where you host ERPNext
- Choose whether to **Ignore SSL Issues**: When selected, n8n will connect even if SSL certificate validation is unavailable.

If you are an ERPNext System Manager, you can also generate API keys and secrets for other users. Refer to the [ERPNext Adding Users documentation](#) for more information.

How to find the subdomain of an ERPNext cloud-hosted account

You can find your ERPNext subdomain by reviewing the address bar of your browser. The string between `https://` and either `.erpnext.com` or `frappe.cloud` is your subdomain.

For example, if the URL in the address bar is `https://n8n.erpnext.com`, the subdomain is `n8n`.

Eventbrite credentials

You can use these credentials to authenticate the following nodes:

- [Eventbrite Trigger](#)

Prerequisites

Create an [Eventbrite](#) account.

Supported authentication methods

- API private key
- OAuth2

Related resources

Refer to [Eventbrite's API documentation](#) for more information about the service.

Using API private key

To configure this credential, you'll need:

- A **Private Key**: Refer to the [Eventbrite API Authentication Get a Private Token documentation](#) for detailed steps to generate a Private Token. Use this private token as the **Private Key** in the n8n credential.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you need to configure OAuth2 from scratch or need more detail on what's happening in the OAuth web flow, refer to the instructions in the [Eventbrite API authentication For App Partners documentation](#) to set up OAuth.

F5 Big-IP credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create an [F5 Big-IP](#) account.

Authentication methods

- Account login

Related resources

Refer to [F5 Big-IP’s API documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n’s website.

Using account login

To configure this credential, you'll need:

- A **Username**: Use the username you use to log in to F5 Big-IP.
 - A **Password**: Use the user password you use to log in to F5 Big-IP.
-

Facebook App credentials

You can use these credentials to authenticate the following nodes:

- [Facebook Trigger](#)

Supported authentication methods

- App access token

Related resources

Refer to [Meta's Graph API documentation](#) for more information about the service.

Using app access token

To configure this credential, you'll need a [Meta for Developers](#) account and:

- An app **Access Token**
- An optional **App Secret**: Used to verify the integrity and origin of the payload.

There are five steps in setting up your credential:

1. [Create a Meta app](#) with the Webhooks product.
2. [Generate an App Access Token](#) for that app.
3. [Configure the Facebook trigger](#).
4. Optional: [Add an app secret](#).
5. [App Review](#): Only required if your app's users don't have roles on the app itself. If you're creating the app for your own internal purposes, this isn't necessary.

Refer to the detailed instructions below for each step.

Create a Meta app

To create a Meta app:

1. Go to the [Meta Developer App Dashboard](#) and select **Create App**.
2. If you have a business portfolio and you're ready to connect the app to it, select the business portfolio. If you don't have a business portfolio or you're not ready to connect the app to the portfolio, select **I don't want to connect a business portfolio yet** and select **Next**. The [Use cases](#) page opens.
3. Select **Other**, then select **Next**.
4. Select **Business** and **Next**.
5. Complete the essential information:
 - Add an **App name**.
 - Add an **App contact email**.
 - Here again you can connect to a business portfolio or skip it.
6. Select **Create app**.
7. The [Add products to your app](#) page opens.
8. Select **App settings > Basic** from the left menu.
9. Enter a **Privacy Policy URL**. (Required to take the app "Live.")
10. Select **Save changes**.
11. At the top of the page, toggle the **App Mode** from **Development** to **Live**.
12. In the left menu, select **Add Product**.
13. The [Add products to your app](#) page appears. Select **Webhooks**.
14. The **Webhooks** product opens.

Refer to Meta's [Create an app](#) documentation for more information on creating an app, required fields like the Privacy Policy URL, and adding products.

For more information on the app modes and switching to **Live** mode, refer to [App Modes](#) and [Publish | App Types](#).

Generate an App Access Token

Next, create an app access token to be used by your n8n credential and the Webhooks product:

1. In a separate tab or window, open the [Graph API explorer](#).
2. Select the **Meta App** you just created in the **Access Token** section.
3. In **User or Page**, select **Get App Token**.
4. Select **Generate Access Token**.
5. The page prompts you to log in and grant access. Follow the on-screen prompts.
6. Copy the token and enter it in your n8n credential as the **Access Token**. Save this token somewhere else, too, since you'll need it for the Webhooks configuration.
7. Save your n8n credential.

Refer to the Meta instructions for [Your First Request](#) for more information on generating the token.

Configure the Facebook Trigger

Now that you have a token, you can configure the Facebook Trigger node:

1. In your Meta app, copy the **App ID** from the top navigation bar.
2. In n8n, open your Facebook Trigger node.
3. Paste the **App ID** into the **APP ID** field.
4. Select **Execute step** to shift the trigger into listening mode.
5. Return to the tab or window where your Meta app's **Webhooks** product configuration is open.
6. **Subscribe** to the objects you want to receive Facebook Trigger notifications about. For each subscription:
 1. Copy the **Webhook URL** from n8n and enter it as the **Callback URL** in your Meta App.
 2. Enter the **Access Token** you copied above as the **Verify token**.
 3. Select **Verify and save**. (This step fails if you don't have your n8n trigger listening.)
 4. Some webhook subscriptions, like **User**, prompt you to subscribe to individual events. Subscribe to the events you're interested in.
 5. You can send some **Test** events from Meta to confirm things are working. If you send a test event, verify its receipt in n8n.

Refer to the [Facebook Trigger node](#) documentation for more information.

Optional: Add an App Secret

For added security, Meta recommends adding an **App Secret**. This signs all API calls with the `appsecret_proof` parameter. The app secret proof is a sha256 hash of your access token, using your app secret as the key.

To generate an App Secret:

1. In Meta while viewing your app, select **App settings > Basic** from the left menu.
2. Select **Show** next to the **App secret** field.
3. The page prompts you to re-enter your Facebook account credentials. Once you do so, Meta shows the App Secret.
4. Highlight it to select it, copy it, and paste this into your n8n credential as the **App Secret**.
5. **Save** your n8n credential.

Refer to the [App Secret documentation](#) for more information.

App review

App Review requires Business Verification.

Your app must go through App Review if it will be used by someone who:

- Doesn't have a role on the app itself.
- Doesn't have a role in the Business that has claimed the app.

If your only app users are users who have a role on the app itself, App Review isn't required.

As part of the App Review process, you may need to request advanced access for your webhook subscriptions.

Refer to Meta's [App Review](#) and [Advanced Access](#) documentation for more information.

Common issues

Unverified apps limit

Facebook only lets you have a developer or administrator role on a maximum of 15 apps that aren't already linked to a Meta Verified Business Account.

Refer to [Limitations | Create an app](#) if you're over that limit.

Facebook Graph API credentials

You can use these credentials to authenticate the following nodes:

- [Facebook Graph API](#)

Supported authentication methods

- App access token

Related resources

Refer to [Meta's Graph API documentation](#) for more information about the service.

Using app access token

To configure this credential, you'll need a [Meta for Developers](#) account and:

- An app **Access Token**

There are two steps in setting up your credential:

1. [Create a Meta app](#) with the products you need to access.
2. [Generate an App Access Token](#) for that app.

Refer to the detailed instructions below for each step.

Create a Meta app

To create a Meta app:

1. Go to the Meta Developer [App Dashboard](#) and select **Create App**.
2. If you have a business portfolio and you're ready to connect the app to it, select the business portfolio. If you don't have a business portfolio or you're not ready to connect the app to the portfolio, select **I don't want to connect a business portfolio yet** and select **Next**. The **Use cases** page opens.
3. Select the **Use case** that aligns with how you wish to use the Facebook Graph API. For example, for products in Meta's **Business** suite (like Messenger, Instagram, WhatsApp, Marketing API, App Events, Audience Network, Commerce API, Fundraisers, Jobs, Threat Exchange, and Webhooks), select **Other**, then select **Next**.
4. Select **Business** and **Next**.
5. Complete the essential information:
 - Add an **App name**.
 - Add an **App contact email**.
 - Here again you can connect to a business portfolio or skip it.
6. Select **Create app**.
7. The **Add products to your app** page opens.
8. Select **App settings > Basic** from the left menu.
9. Enter a **Privacy Policy URL**. (Required to take the app "Live.")
10. Select **Save changes**.
11. At the top of the page, toggle the **App Mode** from **Development** to **Live**.
12. In the left menu, select **Add Product**.
13. The **Add products to your app** page appears. Select the products that make sense for your app and configure them.

Refer to Meta's [Create an app](#) documentation for more information on creating an app, required fields like the Privacy Policy URL, and adding products.

For more information on the app modes and switching to **Live** mode, refer to [App Modes](#) and [Publish | App Types](#).

Generate an App Access Token

Next, create an app access token to use with your n8n credential and the products you selected:

1. In a separate tab or window, open the [Graph API explorer](#).
2. Select the **Meta App** you just created in the **Access Token** section.
3. In **User or Page**, select **Get App Token**.
4. Select **Generate Access Token**.
5. The page prompts you to log in and grant access. Follow the on-screen prompts.
6. Copy the token and enter it in your n8n credential as the **Access Token**.

Refer to the Meta instructions for [Your First Request](#) for more information on generating the token.

Facebook Lead Ads credentials

You can use these credentials to authenticate the following nodes:

- [Facebook Lead Ads trigger](#)

Supported authentication methods

- OAuth2

Related resources

Refer to [Facebook Lead Ads' documentation](#) for more information about the service.

View [example workflows and related content](#) on n8n's website.

Using OAuth2

To configure this credential, you'll need a [Meta for Developers](#) account and:

- A **Client ID**
- A **Client Secret**

To get both, [create a Meta app](#) with either the Facebook Login product or the Facebook Login for Business product.

To create your app and set up the credential with **Facebook Login for Business**:

1. Go to the Meta Developer [App Dashboard](#) and select **Create App**.
2. If you have a business portfolio and you're ready to connect the app to it, select the business portfolio. If you don't have a business portfolio or you're not ready to connect the app to the portfolio, select **I don't want to connect a business portfolio yet** and select **Next**. The **Use cases** page opens.
3. Select **Other**, then select **Next**.
4. Select **Business** and **Next**.
5. Complete the essential information:
 - Add an **App name**.
 - Add an **App contact email**.
 - Here again you can connect to a business portfolio or skip it.
6. Select **Create app**. The **Add products to your app** page opens.
7. Select **Facebook Login for Business**. The **Settings** page for this product opens.
8. Copy the **OAuth Redirect URL** from your n8n credential.
9. In your Meta app settings in **Client OAuth settings**, paste that URL as the **Valid OAuth Redirect URIs**.
10. Select **App settings > Basic** from the left menu.
11. Copy the **App ID** and enter it as the **Client ID** within your n8n credential.
12. Copy the **App Secret** and enter it as the **Client Secret** within your n8n credential.

Your credential should successfully connect now, but you'll need to go through the steps to take your Meta app live before you can use it with the [Facebook Lead Ads trigger](#). Here's a summary of what you'll need to do:

1. In your Meta app, select **App settings > Basic** from the left menu.
2. Enter a **Privacy Policy URL**. (Required to take the app "Live.")
3. Select **Save changes**.
4. At the top of the page, toggle the **App Mode** from **Development** to **Live**.
5. Facebook Login for Business requires Advanced Access for `public_profile`. To add it, go to **App Review > Permissions and Features**.
6. Search for `public_profile` and select **Request advanced access**.
7. Complete the steps for [business verification](#).
8. Use the [Lead Ads Testing Tool](#) to trigger some demo form submissions and test your workflow.

Refer to Meta's [Create an app](#) documentation for more information on creating an app, required fields like the Privacy Policy URL, and adding products.

For more information on the app modes and switching to **Live** mode, refer to [App Modes](#) and [Publish | App Types](#).

Figma credentials

You can use these credentials to authenticate the following nodes:

- [Figma Trigger \(Beta\)](#)

Prerequisites

Create a [Figma](#) account. You need an admin or owner level account.

Supported authentication methods

- API key

Related resources

Refer to [Figma's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- A Personal **Access Token** (PAT): Refer to the [Figma API Access Tokens documentation](#) for instructions on generating a Personal Access Token.
-

FileMaker credentials

You can use these credentials to authenticate the following nodes:

- [FileMaker](#)

Prerequisites

- Create a user account on a [FileMaker Server](#) with the `fmrest` extended privilege to [Access the FileMaker Data API](#).
- Ensure the FileMaker Server can use the [FileMaker Data API](#):
 1. Prepare your database for FileMaker Data API access using FileMaker Pro. You can create a database or prepare an existing database.
 - Refer to [Prepare databases for FileMaker Data API access](#) for more information.
 2. Write code that calls FileMaker Data API methods to find, create, edit, duplicate, and delete records in a hosted database.
 - Refer to [Write FileMaker Data API calls](#) for more information.
 3. Host your solution with FileMaker Data API access enabled.
 - Refer to [Host a FileMaker Data API solution](#) for more information.
 4. Test that FileMaker Data API access is working.
 - Refer to [Test the FileMaker Data API solution](#) for more information.

5. Monitor your hosted solution using Admin Console.
 - Refer to [Monitor FileMaker Data API solutions](#) for more information.

Supported authentication methods

- Database connection

Related resources

Refer to [FileMaker's Data API Guide](#) for more information about the service.

Using database connection

To configure this credential:

1. Enter the **Host** name or IP address of your FileMaker Server.
 2. Enter the **Database** name. This should match the database name as it appears in the **Databases** list within FileMaker.
 3. Enter the user account **Login** for the account with the `fmrest` extended privilege. Refer to the previous [Prerequisites](#) section for more information.
 4. Enter the **Password** for that user account.
-

Filescan credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create a [Filescan](#) account.

Related resources

Refer to [Filescan's API documentation](#) for more information about authenticating with the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows](#) and related content on n8n's website.

Using API key

To configure this credential, you'll need:

- An **API Key**: Generate your API key from your [profile settings](#) > [API Key](#). Refer to the [Filescan FAQ](#) for more information.
-

Flow credentials

You can use these credentials to authenticate the following nodes:

- [Flow](#)
- [Flow Trigger](#)

Prerequisites

Create a [Flow](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Flow's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- Your numeric **Organization ID**
- An **Access Token**

Refer to the [Flow API Getting Started documentation](#) for instructions on generating your Access Token and viewing your Organization ID.

Form.io Trigger credentials

You can use these credentials to authenticate the following nodes:

- [Form.io Trigger](#)

Supported authentication methods

- Basic auth

Related resources

Refer to [Form.io's API documentation](#) for more information about the service.

Using basic auth

To configure this credential, you'll need a [Form.io](#) account and:

- Your **Environment**
- Your login **Email address**
- Your **Password**

To set up the credential:

1. Select your **Environment**:
 - Choose **Cloud hosted** if you aren't hosting Form.io yourself.
 - Choose **Self-hosted** if you're hosting Form.io yourself. Then add:
 - Your **Self-Hosted Domain**. Use only the domain itself. For example, if you view a form at <https://yourserver.com/yourproject/manage/view>, the Self-Hosted Domain is <https://yourserver.com>.
 2. Enter the **Email address** you use to log in to Form.io.
 3. Enter the **Password** you use to log in to Form.io.
-

Formstack Trigger credentials

You can use these credentials to authenticate the following nodes:

- [Formstack Trigger](#)

Prerequisites

Create a [Formstack](#) account.

Supported authentication methods

- API access token
- OAuth2

Related resources

Refer to [Formstack's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need:

- An **API Access Token**: To generate an Access Token, [create a new application](#) in Formstack using the following details:
 - **Redirect URI**: For cloud n8n instances, enter <https://oauth.n8n.cloud/oauth2/callback>.
 - For self-hosted n8n instances, enter the OAuth callback URL for your n8n instance in the format https://<n8n_url>/rest/oauth2-credential/callback. For example <https://localhost:5678/rest/oauth2-credential/callback>.
- **Platform**: Select **Website**.

Once you've created the application, copy the access token either from the applications list or by selecting the application to view its details.

Refer to [Formstack's API Authorization documentation](#) for more detailed instructions.

Using OAuth2

To configure this credential, you'll need:

- A **Client ID**
- A **Client Secret**

To generate both of these, [create a new application](#) in Formstack using the following details:

- **Redirect URI:** Copy the **OAuth Redirect URL** from the n8n credential to enter here.
 - For self-hosted n8n instances, enter the OAuth callback URL for your n8n instance in the format `https://<n8n_url>/rest/oauth2-credential/callback`. For example `https://localhost:5678/rest/oauth2-credential/callback`.
- **Platform:** Select **Website**.

Once you've created the application, select it from the applications list to view the **Application Details**. Copy the **Client ID** and **Client Secret** and add them to n8n. Once you've added both, select the **Connect my account** button to begin the OAuth2 flow and authorization process.

Refer to [Formstack's API Authorization documentation](#) for more detailed instructions.

Fortinet FortiGate credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create a [Fortinet FortiGate](#) account.

Supported authentication methods

- API access token

Related resources

Refer to [Fortinet FortiGate's API documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n's website.

Using API access token

To configure this credential, you'll need:

- An **API Access Token**: To generate an access token, create a [REST API administrator](#).

Refer to the [Fortinet FortiGate Using APIs documentation](#) for more information about token-based authentication in FortiGate.

Freshdesk credentials

You can use these credentials to authenticate the following nodes:

- [Freshdesk](#)

Prerequisites

Create a [Freshdesk](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Freshdesk's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Refer to the [Freshdesk API authentication documentation](#) for detailed instructions on getting your API key.
 - A Freshdesk **Domain**: Use the subdomain of your Freshdesk account. This is part of the URL, for example <https://<subdomain>.freshdesk.com>. So if you access Freshdesk through <https://n8n.freshdesk.com>, enter n8n as your **Domain**.
-

Freshservice credentials

You can use these credentials to authenticate the following nodes:

- [Freshservice](#)

Prerequisites

Create a [Freshservice account](#).

Supported authentication methods

- API key

Related resources

Refer to [Freshservice's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Refer to the [Freshservice API authentication documentation](#) for detailed instructions on getting your API key.
 - Your Freshservice **Domain**: Use the subdomain of your Freshservice account. This is part of the URL, for example <https://<subdomain>.freshservice.com>. So if you access Freshservice through <https://n8n.freshservice.com>, enter n8n as your **Domain**.
-

Freshworks CRM credentials

You can use these credentials to authenticate the following nodes:

- [Freshworks CRM](#)

Prerequisites

Create a [Freshworks CRM account](#).

Supported authentication methods

- API key

Related resources

Refer to [Freshworks CRM's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Refer to the [Freshworks CRM API authentication documentation](#) for detailed instructions on getting your API key.
 - Your Freshworks CRM **Domain**: Use the subdomain of your Freshworks CRM account. This is part of the URL, for example <https://<subdomain>.myfreshworks.com>. So if you access Freshworks CRM through <https://n8n.myfreshworks.com>, enter n8n as your **Domain**.
-

FTP credentials

You can use these credentials to authenticate the following nodes:

- [FTP](#)

Prerequisites

Create an account on a File Transfer Protocol (FTP) server like [JSCAPE](#), [OpenSSH](#), or [FileZilla Server](#).

Supported authentication methods

- **FTP account**: Use this method if your FTP server doesn't support SSH tunneling or encrypted connections.
- **SFTP account**: Use this method if your FTP server supports SSH tunneling and encrypted connections.

Related resources

File Transfer Protocol (FTP) and Secure Shell File Transfer Protocol (SFTP) are protocols for transferring files directly between an FTP/SFTP client and server.

Using FTP account

Use this method if your FTP server doesn't support SSH tunneling or encrypted connections.

To configure this credential, you'll need to:

1. Enter the name or IP address of your FTP server's **Host**.
2. Enter the **Port** number the connection should use.
3. Enter the **Username** the credential should connect as.
4. Enter the user's **Password**.

Review your FTP server provider's documentation for instructions on getting the information you need.

Using SFTP account

Use this method if your FTP server supports SSH tunneling and encrypted connections.

To configure this credential, you'll need to:

1. Enter the name or IP address of your FTP server's **Host**.
2. Enter the **Port** number the connection should use.
3. Enter the **Username** the credential should connect as.
4. Enter the user's **Password**.
5. For the **Private Key**, enter a string for either key-based or host-based user authentication
 - Enter your Private Key in OpenSSH format. This is most often generated using the ssh-keygen -o parameter, for example: ssh-keygen -o -a 100 -t ed25519.
6. If the **Private Key** is encrypted, enter the **Passphrase** used to decrypt it.
 - If the **Private Key** doesn't use a passphrase, leave this field blank.

Review your FTP server provider's documentation for instructions on getting the information you need.

GetResponse credentials

You can use these credentials to authenticate the following nodes:

- [GetResponse](#)
- [GetResponse Trigger](#)

Prerequisites

Create a [GetResponse](#) account.

Supported authentication methods

- API key
- OAuth2

Related resources

Refer to [GetResponse's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: To view or generate an API key, go to **Integrations and API > API**. Refer to the [GetResponse Help Center](#) for more detailed instructions.

Using OAuth2

To configure this credential, you'll need:

- A **Client ID**: Generated when you [register your application](#).
- A **Client Secret**: Generated when you [register your application](#) as the **Client Secret Key**.

When you register your application, copy the **OAuth Redirect URL** from n8n and add it as the **Redirect URL** in GetResponse.

Configure OAuth2 credentials for a local environment

GetResponse doesn't accept the localhost callback URL. Follow the steps below to configure the OAuth credentials for a local environment: 1. Use [ngrok](#) to expose the local server running on port 5678 to the internet. In your terminal, run the following command:

```
ngrok http 5678
```

2. Run the following command in a new terminal. Replace <YOUR-NGROK-URL> with the URL that you got from the previous step.

```
export WEBHOOK_URL=<YOUR-NGROK-URL>
```

3. Follow the [Using OAuth2](#) instructions to configure your credentials, using this URL as your **Redirect URL**.
-

Ghost credentials

You can use these credentials to authenticate the following nodes:

- [Ghost](#)

Prerequisites

Create a [Ghost](#) account.

Supported authentication methods

- Admin API key
- Content API key

The keys are generated following the same steps, but the authorization flows and key format are different, so n8n stores the credentials separately. The Content API uses an API key; the Admin API uses an API key to generate a token for authentication.

Related resources

Refer to Ghost's [Admin API documentation](#) for more information about the Admin API service. Refer to Ghost's [Content API documentation](#) for more information about the Content API service.

Using Admin API key

To configure this credential, you'll need:

- The **URL** of your Ghost admin domain. Your admin domain can be different to your main domain and may include a subdirectory. All Ghost(Pro) blogs have a *.ghost.io domain as their admin domain and require https.
- An **API Key**: To generate a new API key, create a new Custom Integration. Refer to the [Ghost Admin API Token Authentication Key documentation](#) for more detailed instructions. Copy the **Admin API Key** and use this as the **API Key** in the Ghost Admin n8n credential.

Using Content API key

To configure this credential, you'll need:

- The **URL** of your Ghost admin domain. Your admin domain can be different to your main domain and may include a subdirectory. All Ghost(Pro) blogs have a *.ghost.io domain as their admin domain and require https.
- An **API Key**: To generate a new API key, create a new Custom Integration. Refer to the [Ghost Content API Key documentation](#) for more detailed instructions. Copy the **Content API Key** and use this as the **API Key** in the Ghost Content n8n credential.

Git credentials

You can use these credentials to authenticate the following nodes:

- [Git](#)

Prerequisites

Create an account on [GitHub](#), [GitLab](#), or similar platforms for use with [Git](#).

Supported authentication methods

- Basic auth

Related resources

Refer to [Git's documentation](#) for more information about the service.

Using basic auth

To configure this credential, you'll need:

- A **Username** for GitHub, GitLab, or a similar platform
 - A **Password** for GitHub, GitLab, or a similar platform
-

GitHub credentials

You can use these credentials to authenticate the following nodes:

- [GitHub](#)
- [GitHub Trigger](#)
- [GitHub Document Loader](#): this node doesn't support OAuth.

Prerequisites

Create a [GitHub](#) account.

Supported authentication methods

- API access token: Use this method with any GitHub nodes.
- OAuth2: Use this method with [GitHub](#) and [GitHub Trigger](#) nodes only; don't use with [GitHub Document Loader](#).

Related resources

Refer to [GitHub's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need a [GitHub](#) account.

There are two steps to setting up this credential:

1. [Generate a GitHub personal access token](#).
2. [Set up the credential](#).

Refer to the sections below for detailed instructions.

Generate personal access token

To generate your personal access token:

1. If you haven't done so already, verify your email address with GitHub. Refer to [Verifying your email address](#) for more information.
2. Open your GitHub profile [Settings](#).
3. In the left navigation, select [Developer settings](#).
4. In the left navigation, under **Personal access tokens**, select **Tokens (classic)**.
5. Select **Generate new token > Generate new token (classic)**.
6. Enter a descriptive name for your token in the **Note** field, like `n8n integration`.
7. Select the **Expiration** you'd like for the token, or select **No**

expiration.

8. Select **Scopes** for your token. For most of the n8n GitHub nodes, add the repo scope.
 - A token without assigned scopes can only access public information.
 - Refer to
9. Select **Generate token**.
10. Copy the token.

Refer to [Creating a personal access token \(classic\)](#) for more information. Refer to [Scopes for OAuth apps](#) for more information on GitHub scopes.

Set up the credential

Then, in your n8n credential:

1. If you aren't using GitHub Enterprise Server, don't change the **GitHub server URL**.
 - If you're using [GitHub Enterprise Server](#), update **GitHub server** to match the URL for your server.
2. Enter your **User** name as it appears in your GitHub profile.
3. Enter the **Access Token** you generated above.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you're [self-hosting](#) n8n, create a new GitHub [OAuth app](#):

1. Open your GitHub profile [Settings](#).
2. In the left navigation, select [Developer settings](#).
3. In the left navigation, select [OAuth apps](#).
4. Select [New OAuth App](#).
 - If you haven't created an app before, you may see [Register a new application](#) instead. Select it.
5. Enter an [Application name](#), like n8n integration.
6. Enter the [Homepage URL](#) for your app's website.
7. If you'd like, add the optional [Application description](#), which GitHub displays to end-users.
8. From n8n, copy the [OAuth Redirect URL](#) and paste it into the GitHub [Authorization callback URL](#).
9. Select [Register application](#).
10. Copy the [Client ID](#) and [Client Secret](#) this generates and add them to your n8n credential.

Refer to the [GitHub Authorizing OAuth apps documentation](#) for more information on the authorization process.

GitLab credentials

You can use these credentials to authenticate the following nodes:

- [GitLab](#)

- [GitLab Trigger](#)

Supported authentication methods

- API access token
- OAuth2 (Recommended)

Related resources

Refer to [GitLab's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need a [GitLab](#) account and:

- The URL of your **GitLab Server**
- An **Access Token**

To set up the credential:

1. In GitLab, select your avatar, then select **Edit profile**.
2. In the left sidebar, select **Access tokens**.
3. Select **Add new token**.
4. Enter a **Name** for the token, like n8n integration.
5. Enter an **expiry date** for the token. If you don't enter an expiry date, GitLab automatically sets it to 365 days later than the current date.
 - The token expires on that expiry date at midnight UTC.
6. Select the desired **Scopes**. For the [GitLab](#) node, use the `api` scope to easily grant access for all the node's functionality. Or refer to [Personal access token scopes](#) to select scopes for the functions you want to use.
7. Select **Create personal access token**.
8. Copy the access token this creates and enter it in your n8n credential as the **Access Token**.
9. Enter the URL of your **GitLab Server** in your n8n credential.

Refer to GitLab's [Create a personal access token documentation](#) for more information.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you're [self-hosting](#) n8n, you'll need a [GitLab](#) account. Then create a new GitLab application:

1. In GitLab, select your avatar, then select **Edit profile**.
2. In the left sidebar, select **Applications**.
3. Select **Add new application**.
4. Enter a **Name** for your application, like n8n integration.
5. In n8n, copy the **OAuth Redirect URL**. Enter it as the GitLab **Redirect URI**.

6. Select the desired **Scopes**. For the [GitLab](#) node, use the api scope to easily grant access for all the node's functionality. Or refer to [Personal access token scopes](#) to select scopes for the functions you want to use.
7. Select **Save application**.
8. Copy the **Application ID** and enter it as the **Client ID** in your n8n credential.
9. Copy the **Secret** and enter it as the **Client Secret** in your n8n credential.

Refer to GitLab's [Configure GitLab as an OAuth 2.0 authentication identity provider](#) documentation for more information. Refer to the [GitLab OAuth 2.0 identity provider API documentation](#) for more information on OAuth2 and GitLab.

Gong credentials

You can use these credentials to authenticate the following nodes:

- [Gong](#)

Supported authentication methods

- API access token
- OAuth2

Related resources

Refer to [Gong's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need a [Gong](#) account and:

- An **Access Key**
- An **Access Key Secret**

You can create both of these items on the [Gong API Page](#) (you must be a technical administrator in Gong to access this resource).

Refer to [Gong's API documentation](#) for more information about authenticating to the service.

Using OAuth2

To configure this credential, you'll need a [Gong](#) account, a [Gong developer](#) account and:

- A **Client ID**: Generated when you create an Oauth app for Gong.
- A **Client Secret**: Generated when you create an Oauth app for Gong.

If you're self-hosting n8n, you'll need to create an app to configure OAuth2. Refer to Gong's OAuth documentation for more information about setting up OAuth2.

Google credentials

This section contains:

- OAuth2 single service: Create an OAuth2 credential for a specific service node, such as the Gmail node.
- OAuth2 generic: Create an OAuth2 credential for use with custom operations.
- Service Account: Create a Service Account credential for some specific service nodes.
- Google PaLM and Gemini: Get a Google Gemini/Google PaLM API key.

OAuth2 and Service Account

There are two authentication methods available for Google services nodes:

- OAuth2: Recommended because it's more widely available and easier to set up.
- Service Account: Refer to the Google documentation: Understanding service accounts for guidance on when you need a service account.

-8<- “_snippets/integrations/managed-google-oauth.md”

Compatible nodes

Once configured, you can use your credentials to authenticate the following nodes. Most nodes are compatible with OAuth2 authentication. Support for Service Account authentication is limited.

| Node | OAuth | Service Account |
|-----------------------------|--------------------|--------------------|
| <u>Google Ads</u> | :white_check_mark: | :x: |
| <u>Gmail</u> | :white_check_mark: | :warning: |
| <u>Google Analytics</u> | :white_check_mark: | :x: |
| <u>Google BigQuery</u> | :white_check_mark: | :white_check_mark: |
| <u>Google Books</u> | :white_check_mark: | :white_check_mark: |
| <u>Google Calendar</u> | :white_check_mark: | :x: |
| <u>Google Chat</u> | :x: | :white_check_mark: |
| <u>Google Cloud Storage</u> | :white_check_mark: | :x: |

| | | |
|---------------------------------------|--------------------|--------------------|
| <u>Google Contacts</u> | :white_check_mark: | :x: |
| <u>Google Cloud Firestore</u> | :white_check_mark: | :white_check_mark: |
| <u>Google Cloud Natural Language</u> | :white_check_mark: | :x: |
| <u>Google Cloud Realtime Database</u> | :white_check_mark: | :x: |
| <u>Google Docs</u> | :white_check_mark: | :white_check_mark: |
| <u>Google Drive</u> | :white_check_mark: | :white_check_mark: |
| <u>Google Drive Trigger</u> | :white_check_mark: | :white_check_mark: |
| <u>Google Perspective</u> | :white_check_mark: | :x: |
| <u>Google Sheets</u> | :white_check_mark: | :white_check_mark: |
| <u>Google Slides</u> | :white_check_mark: | :white_check_mark: |
| <u>Google Tasks</u> | :white_check_mark: | :x: |
| <u>Google Translate</u> | :white_check_mark: | :white_check_mark: |
| <u>Google Workspace Admin</u> | :white_check_mark: | :x: |
| <u>YouTube</u> | :white_check_mark: | :x: |

Google: OAuth2 single service

This document contains instructions for creating a Google credential for a single service. They're also available as a [video](#).

-8<- “_snippets/integrations/managed-google-oauth.md”

Prerequisites

- Create a [Google Cloud](#) account.

Set up OAuth

There are five steps to connecting your n8n credential to Google services:

1. [Create a Google Cloud Console project.](#)
2. [Enable APIs.](#)
3. [Configure your OAuth consent screen.](#)
4. [Create your Google OAuth client credentials.](#)
5. [Finish your n8n credential.](#)

Create a Google Cloud Console project

First, create a Google Cloud Console project. If you already have a project, jump to the [next section](#):

-8<- “ snippets/integrations/builtin/credentials/google/create-google-cloud-project.md”

Enable APIs

With your project created, enable the APIs you'll need access to:

-8<- “ snippets/integrations/builtin/credentials/google/enable-apis.md”

Configure your OAuth consent screen

If you haven't used OAuth in your Google Cloud project before, you'll need to [configure the OAuth consent screen](#):

1. Access your [Google Cloud Console - Library](#). Make sure you're in the correct project.

[Image: The project dropdown in the Google Cloud top navigation]
Check the project dropdown in the Google Cloud top navigation
2. Open the left navigation menu and go to **APIs & Services > OAuth consent screen**. Google will redirect you to the Google Auth Platform overview page.
3. Select **Get started** on the **Overview** tab to begin configuring OAuth consent.
4. Enter an **App name** and **User support email** to include on the Oauth screen. Select **Next** to continue.
5. For the **Audience**, select **Internal** for user access within your organization's Google workspace or **External** for any user with a Google account. Refer to Google's [User type documentation](#) for more information on user types. Select **Next** to continue.
6. Select the **Email addresses** Google should use to contact you about changes to your project. Select **Next** to continue.
7. Read and accept the Google's User Data Policy. Select **Continue** and then select **Create**.
8. In the left-hand menu, select **Branding**.
9. In the **Authorized domains** section, select **Add domain**:
 - If you're using n8n's Cloud service, add n8n.cloud
 - If you're [self-hosting](#), add the domain of your n8n instance.
10. Select **Save** at the bottom of the page.

Create your Google OAuth client credentials

Next, create the OAuth client credentials in Google:

1. Access your [Google Cloud Console](#). Make sure you're in the correct project.
2. In the **APIs & Services** section, select [**Credentials**](#).
3. Select **+ Create credentials** > **OAuth client ID**.
4. In the **Application type** dropdown, select **Web application**.
5. Google automatically generates a **Name**. Update the **Name** to something you'll recognize in your console.
6. **If you're self-hosting:** From your n8n credential, copy the **OAuth Redirect URL**. Paste it into the **Authorized redirect URIs** in Google Console. If you're using **n8n cloud**, you can leave this field empty as the OAuth setup is pre-configured, and the callback URL is fixed for that configuration.
7. Select **Create**.

Finish your n8n credential

With the Google project and credentials fully configured, finish the n8n credential: - If **self-hosted**:

1. From Google's **OAuth client created** modal, copy the **Client ID**. Enter this in your n8n credential.
 2. From the same Google modal, copy the **Client Secret**. Enter this in your n8n credential.
- **If n8n cloud:**
 1. In n8n, select **Sign in with Google** to complete your Google authentication.
 2. **Save** your new credentials.

Video

Troubleshooting

Google hasn't verified this app

-8<- “_snippets/integrations/builtin/credentials/google/unverified-app.md”

Google Cloud app becoming unauthorized

-8<- “_snippets/integrations/builtin/credentials/google/app-becoming-unauthorized.md”

Google: OAuth2 generic

This document contains instructions for creating a generic OAuth2 Google credential for use with [custom operations](#).

-8<- “_snippets/integrations/managed-google-oauth.md”

Prerequisites

- Create a [Google Cloud](#) account.

Set up OAuth

There are five steps to connecting your n8n credential to Google services:

1. [Create a Google Cloud Console project](#).
2. [Enable APIs](#).
3. [Configure your OAuth consent screen](#).
4. [Create your Google OAuth client credentials](#).
5. [Finish your n8n credential](#).

Create a Google Cloud Console project

First, create a Google Cloud Console project. If you already have a project, jump to the [next section](#):

-8<- “_snippets/integrations/builtin/credentials/google/create-google-cloud-project.md”

Enable APIs

With your project created, enable the APIs you'll need access to:

-8<- “_snippets/integrations/builtin/credentials/google/enable-apis.md”

Configure your OAuth consent screen

If you haven't used OAuth in your Google Cloud project before, you'll need to [configure the OAuth consent screen](#):

1. Access your [Google Cloud Console - Library](#). Make sure you're in the correct project.

[Image: The project dropdown in the Google Cloud top navigation]
Check the project dropdown in the Google Cloud top navigation
2. Open the left navigation menu and go to **APIs & Services > OAuth consent screen**. Google will redirect you to the Google Auth Platform overview page.
3. Select **Get started** on the **Overview** tab to begin configuring OAuth consent.
4. Enter an **App name** and **User support email** to include on the Oauth screen. Select **Next** to continue.
5. For the **Audience**, select **Internal** for user access within your organization's Google workspace or **External** for any user with a Google account. Refer to Google's [User type documentation](#) for more information on user types. Select **Next** to continue.
6. Select the **Email addresses** Google should use to contact you about changes to your project. Select **Next** to continue.
7. Read and accept the Google's User Data Policy. Select **Continue** and then select **Create**.
8. In the left-hand menu, select **Branding**.

9. In the **Authorized domains** section, select **Add domain**:
 - If you're using n8n's Cloud service, add `n8n.cloud`
 - If you're self-hosting, add the domain of your n8n instance.
10. Select **Save** at the bottom of the page.

Create your Google OAuth client credentials

Next, create the OAuth client credentials in Google:

1. Access your [Google Cloud Console](#). Make sure you're in the correct project.
2. In the **APIs & Services** section, select [**Credentials**](#).
3. Select **+ Create credentials > OAuth client ID**.
4. In the **Application type** dropdown, select **Web application**.
5. Google automatically generates a **Name**. Update the **Name** to something you'll recognize in your console.
6. From your n8n credential, copy the **OAuth Redirect URL**. Paste it into the **Authorized redirect URIs** in Google Console.
7. Select **Create**.

Finish your n8n credential

With the Google project and credentials fully configured, finish the n8n credential:

1. From Google's **OAuth client created** modal, copy the **Client ID**. Enter this in your n8n credential.
2. From the same Google modal, copy the **Client Secret**. Enter this in your n8n credential.
3. You must provide the scopes for this credential. Refer to [Scopes](#) for more information. Enter multiple scopes in a space-separated list, for example: `https://www.googleapis.com/auth/gmail.labels https://www.googleapis.com/auth/gmail.addons.current.action.compose`
4. In n8n, select **Sign in with Google** to complete your Google authentication.
5. **Save** your new credentials.

Video

The following video demonstrates the steps described above:

Scopes

Google services have one or more possible access scopes. A scope limits what a user can do. Refer to [OAuth 2.0 Scopes for Google APIs](#) for a list of scopes for all services.

n8n doesn't support all scopes. When creating a generic Google OAuth2 API credential, you can enter scopes from the **Supported scopes** list below. If you enter a scope that n8n doesn't already support, it won't work.

??? Details "Supported scopes" | Service | Available scopes ||

-----|

-----|

| | Gmail |

- <https://www.googleapis.com/auth/gmail.labels>
- <https://www.googleapis.com/auth/gmail.addons.current.action.compose>
- <https://www.googleapis.com/auth/gmail.addons.current.message.action>
- <https://mail.google.com/>
- <https://www.googleapis.com/auth/gmail.modify>
- <https://www.googleapis.com/auth/gmail.compose>

| | Google Ads |

- <https://www.googleapis.com/auth/adwords>

|

| Google Analytics |

<https://www.googleapis.com/auth/analytics>

<https://www.googleapis.com/auth/analytics.readonly>

|

| Google BigQuery |

<https://www.googleapis.com/auth/bigquery>

|

| Google Books |

<https://www.googleapis.com/auth/books>

|

| Google Calendar |

<https://www.googleapis.com/auth/calendar>

<https://www.googleapis.com/auth/calendar.events>

|

| Google Cloud
Natural Language |

<https://www.googleapis.com/auth/cloud-language>

<https://www.googleapis.com/auth/cloud-platform>

|

| Google Cloud
Storage |

<https://www.googleapis.com/auth/cloud-platform>

<https://www.googleapis.com/auth/cloud-platform.read-only>

https://www.googleapis.com/auth/devstorage.full_control

https://www.googleapis.com/auth/devstorage.read_only

https://www.googleapis.com/auth/devstorage.read_write

 |

| Google Contacts |

<https://www.googleapis.com/auth/contacts>

|

| Google Docs |

<https://www.googleapis.com/auth/documents>

<https://www.googleapis.com/auth/drive>

<https://www.googleapis.com/auth/drive.file>

|

| Google Drive |

<https://www.googleapis.com/auth/drive>

<https://www.googleapis.com/auth/drive.appdata>

<https://www.googleapis.com/auth/drive.photos.readonly>

|

| Google Firebase
Cloud Firestore |

<https://www.googleapis.com/auth/datasource>

<https://www.googleapis.com/auth/firebase>

```

|
| Google Firebase<br>Realtime Database | <ul>
<li>`https://www.googleapis.com/auth/userinfo.email`</li>
<li>`https://www.googleapis.com/auth.firebaseio.database`</li>
<li>`https://www.googleapis.com/auth.firebaseio`</li></ul>
|
| Google Perspective | <ul>
<li>`https://www.googleapis.com/auth/userinfo.email`</li></ul>
|
| Google Sheets | <ul>
<li>`https://www.googleapis.com/auth/drive.file`</li>
<li>`https://www.googleapis.com/auth/spreadsheets`</li></ul>
|
| Google Slide | <ul>
<li>`https://www.googleapis.com/auth/drive.file`</li>
<li>`https://www.googleapis.com/auth/presentations`</li></ul>
|
| Google Tasks | <ul>
<li>`https://www.googleapis.com/auth/tasks`</li></ul>
|
| Google Translate | <ul>
<li>`https://www.googleapis.com/auth/cloud-translation`</li></ul>
|
| GSuite Admin | <ul>
<li>`https://www.googleapis.com/auth/admin.directory.group`</li>
<li>`https://www.googleapis.com/auth/admin.directory.user`</li>
<li>`https://www.googleapis.com/auth/admin.directory.domain.readonly`</li>
<li>`https://www.googleapis.com/auth/admin.directory.userschema.readonly`</li>
</ul>
|

```

Troubleshooting

Google hasn't verified this app

-8<- “_snippets/integrations/builtin/credentials/google/unverified-app.md”

Google Cloud app becoming unauthorized

-8<- “_snippets/integrations/builtin/credentials/google/app-becoming-unauthorized.md”

Google: Service Account

Using service accounts is more complex than OAuth2. Before you begin:

- Check if your node is compatible with Service Account.
- Make sure you need to use Service Account. For most use cases, OAuth2 is a better option.
- Read the Google documentation on [Creating and managing service accounts](#).

Prerequisites

- Create a [Google Cloud](#) account.

Set up Service Account

There are four steps to connecting your n8n credential to a Google Service Account:

1. [Create a Google Cloud Console project.](#)
2. [Enable APIs.](#)
3. [Set up Google Cloud Service Account.](#)
4. [Finish your n8n credential.](#)

Create a Google Cloud Console project

First, create a Google Cloud Console project. If you already have a project, jump to the next section:

-8<- “_snippets/integrations/builtin/credentials/google/create-google-cloud-project.md”

Enable APIs

With your project created, enable the APIs you'll need access to:

-8<- “_snippets/integrations/builtin/credentials/google/enable-apis.md”

Set up Google Cloud Service Account

1. Access your [Google Cloud Console - Library](#). Make sure you're in the correct project.

[Image: The project dropdown in the Google Cloud top navigation]

Check the project dropdown in the Google Cloud top navigation

2. Open the left navigation menu and go to **APIs & Services > Credentials**. Google takes you to your **Credentials** page.
3. Select **+ Create credentials > Service account**.
4. Enter a name in **Service account name** and an ID in **Service account ID**. Refer to [Creating a service account](#) for more information.
5. Select **Create and continue**.
6. Based on your use-case, you may want to **Select a role** and **Grant users access to this service account** using the corresponding sections.
7. Select **Done**.

8. Select your newly created service account under the **Service Accounts** section. Open the **Keys** tab.
9. Select **Add key > Create new key**.
10. In the modal that appears, select **JSON**, then select **CREATE**. Google saves the file to your computer.

Finish your n8n credential

With the Google project and credentials fully configured, finish the n8n credential:

1. Open the downloaded JSON file.
2. Copy the `client_email` and enter it in your n8n credential as the **Service Account Email**.
3. Copy the `private_key`. Don't include the surrounding " marks. Enter this as the **Private Key** in your n8n credential.
4. **Optional:** Choose if you want to **Impersonate a User** (turned on).
 1. To use this option, you must Enable domain-wide delegation for the service account as a Google Workspace super admin.
 2. Enter the **Email** of the user you want to impersonate.
5. If you plan to use this credential with the **HTTP Request** node, turn on **Set up for use in HTTP Request node**.
 1. With this setting turned on, you'll need to add **Scope(s)** for the node. n8n prepopulates some scopes. Refer to **OAuth 2.0 Scopes for Google APIs** for more information.
6. **Save** your credentials.

Video

Troubleshooting

Service Account can't access Google Drive files

A Service Account can't access Google Drive files and folders that weren't shared with its associated user email.

1. Access your **Google Cloud Console** and copy your Service Account email.
2. Access your **Google Drive** and go to the designated file or folder.
3. Right-click on the file or folder and select **Share**.
4. Paste your Service Account email into **Add People and groups**.
5. Select **Editor** for read-write access or **Viewer** for read-only access.

Enable domain-wide delegation

To impersonate a user with a service account, you must enable domain-wide delegation for the service account.

To delegate domain-wide authority to a service account, you must be a super administrator for the Google Workspace domain. Then:

1. From your Google Workspace domain's [Admin console](#), select the hamburger menu, then select **Security > Access and data control > API Controls**.
2. In the **Domain wide delegation** pane, select **Manage Domain Wide Delegation**.
3. Select **Add new**.
4. In the **Client ID** field, enter the service account's **Client ID**. To get the Client ID:
 - Open your Google Cloud Console project, then open the [Service Accounts](#) page.
 - Copy the **OAuth 2 Client ID** and use this as the **Client ID** for the **Domain Wide Delegation**.
5. In the **OAuth scopes** field, enter a list of comma-separate scopes to grant your application access. For example, if your application needs domain-wide full access to the Google Drive API and the Google Calendar API, enter:
`https://www.googleapis.com/auth/drive,`
`https://www.googleapis.com/auth/calendar.`
6. Select **Authorize**.

It can take from 5 minutes up to 24 hours before you can impersonate all users in your Workspace.

Google Gemini(PaLM) credentials

You can use these credentials to authenticate the following nodes:

- [Embeddings Google Gemini](#)
- [Google Gemini](#)
- [Google Gemini Chat Model](#)
- [Embeddings Google PaLM](#)

Prerequisites

- Create a [Google Cloud](#) account.
- Create a [Google Cloud Platform](#) project.

Supported authentication methods

- Gemini(PaLM) API key

Related resources

Refer to [Google's Gemini API documentation](#) for more information about the service.

-8<- “snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Using Gemini(PaLM) API key

To configure this credential, you'll need:

- The **API Host URL**: Both PaLM and Gemini use the default <https://generativelanguage.googleapis.com>.
- An **API Key**: Create a key in [Google AI Studio](#).

To create an API key:

1. Go to the API Key page in Google AI Studio: <https://aistudio.google.com/apikey>.
 2. Select **Create API Key**.
 3. You can choose whether to **Create API key in new project** or search for an existing Google Cloud project to **Create API key in existing project**.
 4. Copy the generated API key and add it to your n8n credential.
-

Gotify credentials

You can use these credentials to authenticate the following nodes:

- [Gotify](#)

Prerequisites

Install [Gotify](#) on your server.

Supported authentication methods

- API token

Related resources

Refer to [Gotify's API documentation](#) for more information about the service.

Using API token

To configure this credential, you'll need:

- An **App API Token**: Only required if you'll use this credential to create messages. To generate an App API token, create an application from the **Apps** menu. Refer to [Gotify's Push messages documentation](#) for more information.
 - A **Client API Token**: Required for all actions other than creating messages (such as deleting or retrieving messages). To generate a Client API token, create a client from the **Clients** menu.
 - The **URL** of the Gotify host
-

GoTo Webinar credentials

You can use these credentials to authenticate the following nodes:

- [GoToWebinar](#)

Prerequisites

Create a [GoToWebinar](#) account with [Developer Center](#) access.

Supported authentication methods

- OAuth2

Related resources

Refer to [GoToWebinar's API documentation](#) for more information about authenticating with the service.

Using OAuth2

To configure this credential, you'll need:

- A **Client ID**: Provided once you create an OAuth client
- A **Client Secret**: Provided once you create an OAuth client

Refer to the [Create an OAuth client documentation](#) for detailed instructions on creating an OAuth client. Copy the **OAuth Callback URL** from n8n to use as the **Redirect URI** in your OAuth client. The Client ID and Client secret are provided once you've finished setting up your client.

Grafana credentials

You can use these credentials to authenticate the following nodes:

- [Grafana](#)

Prerequisites

- Create a [Grafana](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Grafana's API documentation](#) for more information about authenticating with the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Refer to the [Create an API key documentation](#) for detailed instructions on creating an API key.
 - The **Base URL** for your Grafana instance, for example: <https://n8n.grafana.net>.
-

Grist credentials

You can use these credentials to authenticate the following nodes:

- [Grist](#)

Prerequisites

Create a [Grist](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Grist's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Refer to the [Grist API authentication documentation](#) for instructions on creating an API key.
 - To select your Grist **Plan Type**. Options include:
 - Free
 - Paid: If selected, provide your Grist **Custom Subdomain**. This is the portion that comes before .getgrist.com. For example, if our full Grist domain was n8n.getgrist.com, we'd enter n8n here.
 - Self-Hosted: If selected, provide your Grist **Self-Hosted URL**. This should be the full URL.
-

Groq credentials

You can use these credentials to authenticate the following nodes:

- [Groq Chat Model](#)

Prerequisites

Create a [Groq](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Groq's documentation](#) for more information about the service.

-8<- “[_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md](#)”

Using API key

To configure this credential, you'll need:

- An **API Key**

To get your API key:

1. Go to the [API Keys](#) page of your Groq console.
2. Select **Create API Key**.
3. Enter a **display name** for the key, like n8n integration, and select **Submit**.
4. Copy the key and paste it into your n8n credential.

Refer to [Groq's API Keys documentation](#) for more information.

Gumroad credentials

You can use these credentials to authenticate the following nodes:

- [Gumroad Trigger](#)

Prerequisites

Create a [Gumroad](#) account.

Supported authentication methods

- API access token

Related resources

Refer to [Gumroad's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need:

- An **API Access Token**: Create an application to generate an access token. Refer to the [Gumroad Create an application for the API documentation](#) for detailed instructions on creating a new application and generating an access token.
-

HaloPSA credentials

You can use these credentials to authenticate the following nodes:

- [HaloPSA](#)

Prerequisites

Create a [HaloPSA](#) account.

Supported authentication methods

- API key

Related resources

Refer to [HaloPSA's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- To select your **Hosting Type**:
 - **On Premise Solution**: Choose this option if you're hosting the Halo application on your own server
 - **Hosted Solution Of Halo**: Choose this option if your application is hosted by Halo. If this option is selected, you'll need to provide your **Tenant**.
- The **HaloPSA Authorisation Server URL**: Your Authorisation Server URL is displayed within HaloPSA in **Configuration > Integrations > Halo API** in [API Details](#).
- The **Resource Server URL**: Your Resource Server is displayed within HaloPSA in **Configuration > Integrations > Halo API** in [API Details](#).
- A **Client ID**: Obtained by registering the application in the Halo API settings. Refer to [HaloPSA's Authorisation documentation](#) for detailed instructions. n8n recommends using these settings:
 - Choose **Client Credentials** as your **Authentication Method**.

- Use the all permission.
- A **Client Secret**: Obtained by registering the application in the Halo API settings.
- Your **Tenant** name: If **Hosted Solution of Halo** is selected as the **Hosting Type**, you must provide your tenant name. Your tenant name is displayed within HaloPSA in **Configuration > Integrations > Halo API** in [API Details](#).

HaloPSA uses both the application permissions and the agent's permissions to determine API access.

Harvest credentials

You can use these credentials to authenticate the following nodes:

- [Harvest](#)

Prerequisites

Create a [Harvest](#) account.

Supported authentication methods

- API access token
- OAuth2

Related resources

Refer to [Harvest's API documentation](#) for more information about the service.

Using API Access Token

To configure this credential, you'll need:

- A Personal **Access Token**: Refer to the [Harvest Personal Access Token Authentication documentation](#) for instructions on creating a personal access token.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you need to configure OAuth2 from scratch or need more detail on what's happening in the OAuth web flow, refer to the instructions in the [Harvest OAuth2 documentation](#) to set up OAuth.

Help Scout credentials

You can use these credentials to authenticate the following nodes:

- [Help Scout](#)
- [Help Scout Trigger](#)

Prerequisites

Create a [Help Scout](#) account.

Supported authentication methods

- OAuth2

Related resources

Refer to [Help Scout's API documentation](#) for more information about the service.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you need to configure OAuth2 from scratch or need more detail on what's happening in the OAuth web flow, you'll need to create a Help Scout app. Refer to the instructions in the [Help Scout OAuth documentation](#) for more information.

HighLevel credentials

You can use these credentials to authenticate the following nodes:

- [HighLevel node](#)

Prerequisites

Create a [HighLevel developer](#) account.

Supported authentication methods

- API key: Use with API v1
- OAuth2: Use with API v2

Related resources

Refer to [HighLevel's API 2.0 documentation](#) for more information about the service.

For existing integrations with the API v1.0, refer to [HighLevel's API 1.0 documentation](#).

Using API key

To configure this credential, you'll need:

- An **API Key**: Refer to the [HighLevel API 1.0 Welcome documentation](#) for instructions on getting your API key.

Using OAuth2

To configure this credential, you'll need:

- A **Client ID**
- A **Client Secret**

To generate both, create an app in **My Apps > Create App**. Use these settings:

1. Set **Distribution Type** to **Sub-Account**.
2. Add these **Scopes**:
 - locations.readonly
 - contacts.readonly
 - contacts.write
 - opportunities.readonly
 - opportunities.write
 - users.readonly
3. Copy the **OAuth Redirect URL** from n8n and add it as a **Redirect URL** in your HighLevel app.
4. Copy the **Client ID** and **Client Secret** from HighLevel and add them to your n8n credential.
5. Add the same scopes added above to your n8n credential in a space-separated list. For example:

```
locations.readonly contacts.readonly contacts.write  
opportunities.readonly opportunities.write users.readonly
```

Refer to HighLevel's [API Authorization documentation](#) for more details. Refer to HighLevel's [API Scopes documentation](#) for more information about available scopes.

Home Assistant credentials

You can use these credentials to authenticate the following nodes:

- [Home Assistant](#)

Supported authentication methods

- API access token

Related resources

Refer to [Home Assistant's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need to [Install Home Assistant](#), create a [Home Assistant](#) account, and have:

- Your **Host**
- The **Port**
- A Long-Lived **Access Token**

To generate an access token and set up the credential:

1. To generate your **Access Token**, log in to Home Assistant and open your [User profile](#).
2. In the **Long-Lived Access Tokens** section, generate a new token.
3. Copy this token and enter it in n8n as your **Access Token**.
4. Enter the URL or IP address of your Home Assistant **Host**, without the `http://` or `https://` protocol, for example `your.awesome.home`.
5. For the **Port**, enter the appropriate port:
 - If you've made no port changes and access Home Assistant at `http://`, keep the default of 8123.
 - If you've made no port changes and access Home Assistant at `https://`, enter 443.
 - If you've configured Home Assistant to use a specific port, enter that port.
6. If you've enabled SSL in Home Assistant in the [config.yml map key](#), turn on the **SSL** toggle in n8n. If you're not sure, it's best to turn this setting on if you access your home assistant UI using `https://` instead of `http://`.

HTTP Request credentials

You can use these credentials to authenticate the following nodes:

- [HTTP Request](#)
- [HTTP Request Tool \(legacy\)](#)

Prerequisites

You must use the authentication method required by the app or service you want to query.

If you need to secure the authentication with an SSL certificate, refer to [Provide an SSL certificate](#) for the information you'll need.

Supported authentication methods

- Predefined credential type
- Basic auth (generic credential type)
- Custom auth (generic credential type)
- Digest auth (generic credential type)
- Header auth (generic credential type)
- Bearer auth (generic credential type)
- OAuth1 (generic credential type)
- OAuth2 (generic credential type)
- Query auth (generic credential type)

Refer to [HTTP authentication](#) for more information relating to generic credential types.

Using predefined credential type

-8<- “_snippets/integrations/predefined-credential-type-how-to.md”

Refer to [Custom API operations](#) for more information.

-8<- “_snippets/integrations/builtin/credentials/generic-auth/basic-auth.md”

-8<- “_snippets/integrations/builtin/credentials/generic-auth/digest-auth.md”

-8<- “_snippets/integrations/builtin/credentials/generic-auth/header-auth.md”

-8<- “_snippets/integrations/builtin/credentials/generic-auth/bearer-auth.md”

Using OAuth1

Use this generic authentication if your app or service supports OAuth1 authentication.

To configure this credential, enter:

- An **Authorization URL**: Also known as the Resource Owner Authorization URI. This URL typically ends in /oauth1/authorize. The temporary credentials are sent here to prompt a user to complete authorization.
- An **Access Token URL**: This is the URI used for the initial request for temporary credentials. This URL typically ends in /oauth1/request or /oauth1/token.
- A **Consumer Key**: Also known as the client key, like a username. This specifies the oauth_consumer_key to use for the call.
- A **Consumer Secret**: Also known as the client secret, like a password.
- A **Request Token URL**: This is the URI used to switch from temporary credentials to long-lived credentials after authorization. This URL typically ends in /oauth1/access.
- Select the **Signature Method** the auth handshake uses. This specifies the oauth_signature_method to use for the call. Options include:
 - **HMAC-SHA1**

- **HMAC-SHA256**
- **HMAC-SHA512**

For most OAuth1 integrations, you'll need to configure an app, service, or integration to generate the values for most of these fields. Use the **OAuth Redirect URL** in n8n as the redirect URL or redirect URI for such a service.

Read more about [OAuth1](#) and the [OAuth1 authorization flow](#).

Using OAuth2

Use this generic authentication if your app or service supports OAuth2 authentication.

Requirements to configure this credential depend on the **Grant Type** selected. Refer to [OAuth Grant Types](#) for more information on each grant type.

For most OAuth2 integrations, you'll need to configure an app, service, or integration. Use the **OAuth Redirect URL** in n8n as the redirect URL or redirect URI for such a service.

Read more about [OAuth2](#).

Authorization Code grant type

Use Authorization Code grant type to exchange an authorization code for an access token. The auth flow uses the redirect URL to return the user to the client. Then the application gets the authorization code from the URL and uses it to request an access token. Refer to [Authorization Code Request](#) for more information.

To configure this credential, select **Authorization Code** as the **Grant Type**.

Then enter:

- An **Authorization URL**
- An **Access Token URL**
- A **Client ID**: The ID or username to log in with.
- A **Client Secret**: The secret or password used to log in with.
- *Optional*: Enter one or more **Scopes** for the credential. If unspecified, the credential will request all scopes available to the client.
- *Optional*: Some services require more query parameters. If your service does, add them as **Auth URI Query Parameters**.
- An **Authentication** type: Select the option that best suits your use case. Options include:
 - **Header**: Send the credentials as a basic auth header.
 - **Body**: Send the credentials in the body of the request.
- *Optional*: Choose whether to **Ignore SSL Issues**. If turned on, n8n will connect even if SSL validation fails.

Client Credentials grant type

Use the Client Credentials grant type when applications request an access token to access their own resources, not on behalf of a user. Refer to [Client Credentials](#) for more information.

To configure this credential, select **Client Credentials** as the **Grant Type**.

Then enter:

- An **Access Token URL**: The URL to hit to begin the OAuth2 flow. Typically this URL ends in /token.
- A **Client ID**: The ID or username to use to log in to the client.
- A **Client Secret**: The secret or password used to log in to the client.
- *Optional:* Enter one or more **Scopes** for the credential. Most services don't support scopes for Client Credentials grant types; only enter scopes here if yours does.
- An **Authentication** type: Select the option that best suits your use case. Options include:
 - **Header**: Send the credentials as a basic auth header.
 - **Body**: Send the credentials in the body of the request.
- *Optional:* Choose whether to **Ignore SSL Issues**. If turned on, n8n will connect even if SSL validation fails.

PKCE grant type

Proof Key for Code Exchange (PKCE) grant type is an extension to the Authorization Code flow to prevent CSRF and authorization code injection attacks.

To configure this credential, select **PKCE** as the **Grant Type**.

Then enter:

- An **Authorization URL**
- An **Access Token URL**
- A **Client ID**: The ID or username to log in with.
- A **Client Secret**: The secret or password used to log in with.
- *Optional:* Enter one or more **Scopes** for the credential. If unspecified, the credential will request all scopes available to the client.
- *Optional:* Some services require more query parameters. If your service does, add them as **Auth URI Query Parameters**.
- An **Authentication** type: Select the option that best suits your use case. Options include:
 - **Header**: Send the credentials as a basic auth header.
 - **Body**: Send the credentials in the body of the request.
- *Optional:* Choose whether to **Ignore SSL Issues**. If turned on, n8n will connect even if SSL validation fails.

Using query auth

Use this generic authentication if your app or service supports passing authentication as a single key/value query parameter. (For multiple query parameters, use [Custom Auth](#).)

To configure this credential, enter:

- A query parameter key or **Name**

- A query parameter **Value**

Using custom auth

Use this generic authentication if your app or service supports passing authentication as multiple key/value query parameters or you need more flexibility than the other generic auth options.

The **Custom Auth** credential expects JSON data to define your credential. You can use headers, qs, body or a mix. Review the examples below to get started.

Sending two headers

```
{
  "headers": {
    "X-AUTH-USERNAME": "username",
    "X-AUTH-PASSWORD": "password"
  }
}
```

Body

```
{
  "body" : {
    "user": "username",
    "pass": "password"
  }
}
```

Query string

```
{
  "qs": {
    "appid": "123456",
    "apikey": "my-api-key"
  }
}
```

Sending header and query string

```
{
  "headers": {
    "api-version": "202404"
  },
  "qs": {
    "apikey": "my-api-key"
  }
}
```

Provide an SSL certificate

You can send an SSL certificate with your HTTP request. Create the SSL certificate as a separate credential for use by the node:

1. In the HTTP Request node **Settings**, turn on **SSL Certificates**.

2. On the **Parameters** tab, add an existing SSL Certificate credential to **Credential for SSL Certificates** or create a new one.

To configure your SSL Certificates credential, you'll need to add:

- The Certificate Authority **CA** bundle
- The **Certificate** (CRT): May also appear as a Public Key, depending on who your issuing CA was and how they format the cert
- The **Private Key** (KEY)
- *Optional:* If the **Private Key** is encrypted, enter a **Passphrase** for the private key.

If your SSL certificate is in a single file (such as a .pfx file), you'll need to open the file to copy details from it to paste into the appropriate fields:

- Enter the Public Key/CRT as the **Certificate**
 - Enter the **Private Key**/KEY in that field
-

HubSpot credentials

You can use these credentials to authenticate the following nodes:

- [HubSpot](#)
- [HubSpot Trigger](#)

Supported authentication methods

- App token: Use with the [HubSpot](#) node.
- Developer API key: Use with the [HubSpot Trigger](#) node.
- OAuth2: Use with the [HubSpot](#) node.

Related resources

Refer to [HubSpot's API documentation](#) for more information about the service. The [HubSpot Trigger](#) node uses the Webhooks API; refer to [HubSpot's Webhooks API documentation](#) for more information about that service.

Using App token

To configure this credential, you'll need a [HubSpot](#) account or [HubSpot developer](#) account and:

- An **App Token**

To generate an app token, create a private app in HubSpot:

1. In your HubSpot account, select the **settings icon** in the main navigation bar.
2. In the left sidebar menu, go to **Integrations > Private Apps**.
3. Select **Create private app**.
4. On the **Basic Info** tab, enter your app's **Name**.
5. Hover over the **placeholder logo** and select the upload icon to

- upload a square image that will serve as the logo for your app.
6. Enter a **Description** for your app.
 7. Open the **Scopes** tab and add the appropriate scopes. Refer to [Required scopes for HubSpot node](#) for a complete list of scopes you should add.
 8. Select **Create app** to finish the process.
 9. In the modal, review the info about your app's access token, then select **Continue creating**.
 10. Once your app's created, open the **Access token card** and select **Show token** to reveal the token.
 11. Copy this token and enter it in your n8n credential.

Refer to the [HubSpot Private Apps documentation](#) for more information.

Using Developer API key

To configure this credential, you'll need a [HubSpot developer account](#) and:

- A **Client ID**: Generated once you create a public app.
- A **Client Secret**: Generated once you create a public app.
- A **Developer API Key**: Generated from your Developer Apps dashboard.
- An **App ID**: Generated once you create a public app.

To create the public app and set up the credential:

1. Log into your [HubSpot app developer account](#).
2. Select **Apps** from the main navigation bar.
3. Select **Get HubSpot API key**. You may need to select the option to **Show key**.
4. Copy the key and enter it in n8n as the **Developer API Key**.
5. Still on the HubSpot **Apps** page, select **Create app**.
6. On the **App Info** tab, add an **App name**, **Description**, **Logo**, and any support contact info you want to provide. Anyone encountering the app would see these.
7. Open the **Auth** tab.
8. Copy the **App ID** and enter it in n8n.
9. Copy the **Client ID** and enter it in n8n.
10. Copy the **Client Secret** and enter it in n8n.
11. In the **Scopes** section, select **Add new scope**.
12. Add all the scopes listed in [Required scopes for HubSpot Trigger node](#) to your app.
13. Select **Update**.
14. Copy the n8n **OAuth Redirect URL** and enter it as the **Redirect URL** in your HubSpot app.
15. Select **Create app** to finish creating the HubSpot app.

Refer to the [HubSpot Public Apps documentation](#) for more detailed instructions.

Required scopes for HubSpot Trigger node

If you're creating an app for use with the [HubSpot Trigger node](#), n8n recommends starting with these scopes:

| Element | Object | Permission | Scope name |
|---------|--------|------------|------------|
|---------|--------|------------|------------|

| | | | |
|-----|-------------------|------|----------------------------|
| n/a | n/a | n/a | oauth |
| CRM | Companies | Read | crm.objects.companies.read |
| CRM | Companies schemas | Read | crm.schemas.companies.read |
| CRM | Contacts | Read | crm.objects.contacts.read |
| CRM | Contacts schemas | Read | crm.schemas.contacts.read |
| CRM | Deals | Read | crm.objects.deals.read |
| CRM | Deals schemas | Read | crm.schemas.deals.read |

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you’re self-hosting n8n, you’ll need to configure OAuth2 from scratch by creating a new public app:

1. Log into your HubSpot app developer account.
2. Select **Apps** from the main navigation bar.
3. Select **Create app**.
4. On the **App Info** tab, add an **App name**, **Description**, **Logo**, and any support contact info you want to provide. Anyone encountering the app would see these.
5. Open the **Auth** tab.
6. Copy the **App ID** and enter it in n8n.
7. Copy the **Client ID** and enter it in n8n.
8. Copy the **Client Secret** and enter it in n8n.
9. In the **Scopes** section, select **Add new scope**.
10. Add all the scopes listed in Required scopes for HubSpot node to your app.
11. Select **Update**.
12. Copy the n8n **OAuth Redirect URL** and enter it as the **Redirect URL** in your HubSpot app.
13. Select **Create app** to finish creating the HubSpot app.

Refer to the HubSpot Public Apps documentation for more detailed instructions. If you need more detail on what’s happening in the OAuth web flow, refer to the HubSpot Working with OAuth documentation.

Required scopes for HubSpot node

If you’re creating an app for use with the HubSpot node, n8n recommends starting with these scopes:

| Element | Object | Permission | Scope name(s) |
|---------|-----------|---------------|---|
| n/a | n/a | n/a | oauth |
| n/a | n/a | n/a | forms |
| n/a | n/a | n/a | tickets |
| CRM | Companies | Read
Write | crm.objects.companies.read
crm.objects.companies.write |

| | | | |
|-----|-------------------|---------------|---|
| CRM | Companies schemas | Read | crm.schemas.companies.read |
| CRM | Contacts schemas | Read | crm.schemas.contacts.read |
| CRM | Contacts | Read
Write | crm.objects.contacts.read
crm.objects.contacts.write |
| CRM | Deals | Read
Write | crm.objects.deals.read
crm.objects.deals.write |
| CRM | Deals schemas | Read | crm.schemas.deals.read |
| CRM | Owners | Read | crm.objects.owners.read |
| CRM | Lists | Write | crm.lists.write |

Hugging Face credentials

You can use these credentials to authenticate the following nodes:

- [Hugging Face Inference](#)
- [Embeddings Hugging Face Inference](#)

Supported authentication methods

- API key

Related resources

Refer to [Hugging Face's documentation](#) for more information about the service.

-8<- “snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Using API key

To configure this credential, you'll need a [Hugging Face](#) account and:

- An **API Key**: Hugging Face calls these API tokens.

To get your API token:

1. Open your Hugging Face profile and go to the **Tokens** section.
2. Copy the token listed there. It should begin with hf_.
3. Enter this API token as your n8n credential **API Key**.

Refer to [Get your API token](#) for more information.

Human AI credentials

You can use these credentials to authenticate the following nodes:

- [Humantic AI](#)

Prerequisites

Create a [Humantic AI](#) account.

You can also try out an API key as a free trial at the [Humantic AI API](#) page.

Supported authentication methods

- API key

Related resources

Refer to [Humantic AI's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Get an API key from the [Humantic AI API](#) page.
-

Hunter credentials

You can use these credentials to authenticate the following nodes:

- [Hunter](#)

Prerequisites

Create a [Hunter](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Hunter's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Generate an API key from your profile in the [dashboard](#). Refer to the [Hunter API Authentication documentation](#) for more information.
-

Hybrid Analysis credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create a [Hybrid Analysis](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Hybrid Analysis' API documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n's website.

Using API key

To configure this credential, you'll need:

- An **API Key**: Refer to the [Hybrid Analysis' API documentation](#) for instructions on generating an API key.
-

IMAP credentials

You can use these credentials to authenticate the following nodes:

- [IMAP Email](#)

Prerequisites

Create an email account on a service with IMAP support.

Supported authentication methods

- User account

Related resources

Internet Message Access Protocol (IMAP) is a standard protocol for receiving email. Most email providers offer instructions on setting up their service with IMAP; refer to your provider's IMAP instructions.

Using user account

To configure this credential, you'll need:

- A **User** name: The email address you're retrieving email for.
- A **Password**: Either the password you use to check email or an app password. Your provider will tell you whether to use your own password or to generate an app password.
- A **Host**: The IMAP host address for your email provider, often formatted as `imap.<provider>.com`. Check with your provider.
- A **Port** number: The default is port 993. Use this port unless your provider or email administrator tells you to use something different.

Choose whether to use **SSL/TLS** and whether to **Allow Self-Signed Certificates**.

Provider instructions

Refer to the quickstart guides for these common email providers.

Gmail

Refer to [Gmail](#).

Outlook.com

Refer to [Outlook.com](#).

Yahoo

Refer to [Yahoo](#).

My provider isn't listed

If your email provider isn't listed here, search for their IMAP settings or IMAP instructions.

Gmail IMAP credentials

Follow these steps to configure the IMAP credentials with a Gmail account.

Prerequisites

To follow these instructions, you must first:

1. [Enable 2-step Verification](#) on your Gmail account.
2. [Generate an app password](#).

Enable 2-step Verification

-8<- “_snippets/integrations/builtin/credentials/email/gmail-two-step-verification.md”

Generate an app password

-8<- “_snippets/integrations/builtin/credentials/email/gmail-app-password.md”

Set up the credential

To set up the IMAP credential with a Gmail account, use these settings:

1. Enter your Gmail email address as the **User**.
2. Enter the app password you generated above as the **Password**.
3. Enter `imap.gmail.com` as the **Host**.
4. For the **Port**, keep the default port number of 993. Check with your email administrator if this port doesn't work.
5. Turn on the **SSL/TLS** toggle.
6. Check with your email administrator about whether to **Allow Self-Signed Certificates**.

Refer to [Add Gmail to another client](#) for more information. You may need to **Enable IMAP** if you're using a personal Google account before June 2024.

Outlook.com IMAP credentials

Follow these steps to configure the IMAP credentials with an Outlook.com account.

Set up the credentials

To set up the IMAP credential with Outlook.com account, use these settings:

1. Enter your Outlook.com email address as the **User**.
2. Enter your Outlook.com password as the **Password**.
3. Enter `outlook.office365.com` as the **Host**.
4. For the **Port**, keep the default port number of 993.
5. Turn on the **SSL/TLS** toggle.
6. Check with your email administrator about whether to **Allow Self-Signed Certificates**.

Refer to Microsoft's [POP, IMAP, and SMTP settings for Outlook.com](#) documentation for more information.

Connection errors

You may receive a connection error if you configured your Outlook.com account as IMAP in multiple email clients. Microsoft is working on a fix for this. For now, try this workaround:

1. Go to [account.live.com/activity](#) and sign in using the email address and password of the affected account.
2. Under **Recent activity**, find the **Session Type** event that matches the most recent time you received the connection error. Select it to expand the details.
3. Select **This was me** to approve the IMAP connection.
4. Retest your n8n credential.

Refer to [What is the Recent activity page?](#) for more information on using this page.

The source for these instructions is [Outlook.com IMAP connection errors](#). Refer to that documentation for more information.

Use an app password

-8<- “_snippets/integrations/builtin/credentials/email/outlook-app-password.md”

Yahoo IMAP credentials

Follow these steps to configure the IMAP credentials with a Yahoo account.

Prerequisites

To follow these instructions, you must first generate an app password:

-8<- “_snippets/integrations/builtin/credentials/email/yahoo-app-password.md”

Set up the credential

To set up the IMAP credential with a Yahoo Mail account, use these settings:

1. Enter your Yahoo email address as the **User**.
2. Enter the app password you generated above as the **Password**.
3. Enter `imap.mail.yahoo.com` as the **Host**.
4. Keep the default **Port** number of 993. Check with your email administrator if this port doesn't work.
5. Turn on the **SSL/TLS** toggle.
6. Check with your email administrator about whether to **Allow Self-**

Signed Certificates.

Refer to [Set up IMAP for Yahoo mail account](#) for more information.

Imperva WAF credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create an [Imperva WAF](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Imperva WAF’s documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n’s website.

Using API key

To configure this credential, you’ll need:

- An **API ID**
- An **API Key**

Refer to [Imperva WAF’s API Key Management documentation](#) for instructions on generating and viewing API Keys and IDs.

Intercom credentials

You can use these credentials to authenticate the following nodes:

- [Intercom](#)

Prerequisites

- Create an [Intercom](#) developer account.
- [Create an app](#) in your developer hub.

Supported authentication methods

- API key

Related resources

Refer to [Intercom's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Intercom automatically generates an **Access Token** when you [create an app](#). Use this **Access Token** as your n8n **API Key**. Refer to [How to get your Access Token](#) for more detailed instructions.
-

Invoice Ninja credentials

You can use these credentials to authenticate the following nodes:

- [Invoice Ninja](#)
- [Invoice Ninja Trigger](#)

Prerequisites

Create an [Invoice Ninja](#) account. Only the Pro and Enterprise plans support API integrations.

Supported authentication methods

- API key

Related resources

Refer to Invoice Ninja's [v4 API documentation](#) and [v5 API documentation](#) for more information about the APIs.

Using API key

To configure this credential, you'll need:

- A **URL**: If Invoice Ninja hosts your installation, use either of the default URLs mentioned. If you're self-hosting your installation, use the URL of your Invoice Ninja instance.
 - An **API Token**: Generate an API token in **Settings > Account Management > API Tokens**.
 - An optional **Secret**, available only for v5 API users
-

Iterable credentials

You can use these credentials to authenticate the following nodes:

- [Iterable](#)

Prerequisites

Create an [Iterable](#) account.

Supported authentication methods

- API key

Related resources

Refer to Iterable's API documentation for more information about the service:

- [US-based Iterable projects](#)
- [Europe-based Iterable projects](#)

Using API key

To configure this credential, you'll need:

- **An API Key:** Refer to [Iterable's Creating API keys documentation](#) for instructions on creating API keys.
-

Jenkins credentials

You can use these credentials to authenticate the following nodes:

- [Jenkins](#)

Prerequisites

Create an account on a [Jenkins](#) instance.

Supported authentication methods

- API token

Related resources

Jenkins doesn't provide public API documentation; API documentation for each page is available from the user interface in the bottom right. Refer to those detailed pages for more information about the service.

Refer to [Jenkins Remote Access API](#) for information on the API and API wrappers.

Using API token

To configure this credential, you'll need:

- The **Jenkins Username**: For the user whom the token belongs to
- A **Personal API Token**: Generate this from the user's [profile details > Configure > Add new token](#). Refer to [these Stack Overflow instructions](#) for more detail.
- The **Jenkins Instance URL**

Jenkins rebuilt their API token setup in 2018. If you're working with an older Jenkins instance, be sure you're using a non-legacy API token. Refer to [Security Hardening: New API token system in Jenkins 2.129+](#) for more information.

Jina AI credentials

You can use these credentials to authenticate the following nodes:

- [Jina AI](#)

Supported authentication methods

- API key

Related resources

Refer to [Jina AI's reader API documentation](#) and [Jina AI's search API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- **API key**: A Jina AI API key. You can get your free API key without creating an account by doing the following:
 1. Visit the [Jina AI website](#).
 2. Select **API** on the page.
 3. Select **API KEY & BILLING** in the API app widget.
 4. Copy the key labeled "This is your unique key. Store it securely!".

Jina AI API keys start with 10 million free tokens that you can use non-commercially. To top up your key or use commercially, scroll on the **API KEY & BILLING** tab of the **API** widget and select the top up option that best fits your needs.

Jira credentials

You can use these credentials to authenticate the following nodes:

- [Jira](#)
- [Jira Trigger](#)

Prerequisites

Create a [Jira Software Cloud](#) or [Server](#) account.

Supported authentication methods

- [SW Cloud API token](#): Use this method with [Jira Software Cloud](#).
- [SW Server account](#): Use this method with [Jira Software Server](#).

Related resources

Refer to [Jira's API documentation](#) for more information about the service.

Using SW Cloud API token

To configure this credential, you'll need an account on [Jira Software Cloud](#).

Then:

1. Log in to your Atlassian profile > **Security > API tokens** page, or jump straight there using this [link](#).
2. Select **Create API Token**.
3. Enter a good **Label** for your token, like `n8n integration`.
4. Select **Create**.
5. Copy the API token.
6. In n8n, enter the **Email** address associated with your Jira account.
7. Paste the API token you copied as your **API Token**.
8. Enter the **Domain** you access Jira on, for example
`https://example.atlassian.net`.

Refer to [Manage API tokens for your Atlassian account](#) for more information.

Using SW Server account

To configure this credential, you'll need an account on [Jira Software Server](#).

Then:

1. Enter the **Email** address associated with your Jira account.
2. Enter your Jira account **Password**.
3. Enter the **Domain** you access Jira on.

JotForm credentials

You can use these credentials to authenticate the following nodes:

- [JotForm Trigger](#)

Supported authentication methods

- API key

Related resources

Refer to [JotForm's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need a [JotForm](#) account and:

- An **API Key**
- The **API Domain**

To set it up:

1. Go to **Settings > API**.
2. Select **Create New Key**.
3. Select the **Name** in JotForm to update the API key name to something meaningful, like n8n integration.
4. Copy the **API Key** and enter it in your n8n credential.
5. In n8n, select the **API Domain** that applies to you based on the forms you're using:
 - **api.jotform.com**: Use this unless the other form types apply to you.
 - **eu-api.jotform.com**: Select this if you're using JotForm [EU Safe Forms](#).
 - **hipaa-api.jotform.com**: Select this if you're using JotForm [HIPAA forms](#).

Refer to the [JotForm API documentation](#) for more information on creating keys and API domains.

JWT credentials

You can use these credentials to authenticate the following nodes:

- [JWT](#)
- [Webhook](#)

Supported authentication methods

- Passphrase: Signed with a secret with HMAC algorithm
- Private key (PEM key): For use with [Private Key JWT](#) with RSA or

ECDSA algorithm

Related resources

Refer to the [JSON Web Token spec](#) for more details.

For a more verbose introduction, refer to the [JWT website](#). Refer to [Introduction to JSON Web Tokens](#), [JSON Web Token \(JWT\)](#) and [Signing Algorithms Overview](#) for more information on selecting between the two types and the algorithms involved.

Using Passphrase

To configure this credential:

1. Select the **Key Type** of **Passphrase**.
2. Enter the Passphrase **Secret**
3. Select the **Algorithm** used to sign the assertion. Refer to [Available algorithms](#) below for a list of supported algorithms.

Using private key (PEM key)

To configure this credential: 1. Select the **Key Type** of **PEM Key**. 2. A **Private Key**: Obtained from generating a Key Pair. Refer to [Generate RSA Key Pair](#) for an example. 3. A **Public Key**: Obtained from generating a Key Pair. Refer to [Generate RSA Key Pair](#) for an example. 4. Select the **Algorithm** used to sign the assertion. Refer to [Available algorithms](#) below for a list of supported algorithms.

Available algorithms

This n8n credential supports the following algorithms:

- HS256
 - HS384
 - HS512
 - RS256
 - RS384
 - RS512
 - ES256
 - ES384
 - ES512
 - PS256
 - PS384
 - PS512
 - none
-

Kafka credentials

You can use these credentials to authenticate the following nodes:

- [Kafka](#)

- [Kafka Trigger](#)

Supported authentication methods

- Client ID

Related resources

Refer to [Kafka's documentation](#) for more information about using the service.

If you're new to Kafka, refer to the [Apache Kafka Quickstart](#) for initial setup.

Refer to [Encryption and Authentication using SSL](#) for working with SSL in Kafka.

Using client ID

To configure this credential, you'll need a running Kafka environment and:

- A **Client ID**
- A list of relevant **Brokers**
- Username/password authentication details if your Kafka environment uses authentication

To set it up:

1. Enter the **CLIENT-ID** of the client or consumer group in the **Client ID** field in your credential.
2. Enter a comma-separated list of relevant **Brokers** for the credential to use in the format <broker-service-name>:<port>. Use the name you gave the broker when you defined it in the services list. For example, kafka-1:9092,kafka-2:9092 would add the brokers kafka-1 and kafka-2 on port 9092.
3. If your Kafka environment doesn't use SSL, turn off the **SSL** toggle.
4. If you've enabled authentication using SASL in your Kafka environment, turn on the **Authentication** toggle. Then add:
 1. The **Username**
 2. The **Password**
 3. Select the broker's configured **SASL Mechanism**. Refer to [SASL configuration](#) for more information. Options include:
 - Plain
 - scram-sha-256
 - scram-sha-512

Keap credentials

You can use these credentials to authenticate the following nodes:

- [Keap](#)
- [Keap Trigger](#)

Prerequisites

Create a [Keap developer account](#).

Supported authentication methods

- OAuth2

Related resources

Refer to Keap's [REST API documentation](#) for more information about the service.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you need to configure OAuth2 from scratch or need more detail on what's happening in the OAuth web flow, refer to the instructions in the [Getting Started with OAuth2 documentation](#).

Kibana credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

- Create an [Elasticsearch](#) account.
- If you're creating a new account to test with, load some sample data into Kibana. Refer to the [Kibana quick start](#) for more information.

Supported authentication methods

- Basic auth

Related resources

Refer to [Kibana's API documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n's website.

Using basic auth

To configure this credential, you'll need:

- The **URL** you use to access Kibana, for example `http://localhost:5601`
 - A **Username**: Use the same username that you use to log in to Elastic.
 - A **Password**: Use the same password that you use to log in to Elastic.
-

Kitemaker credentials

You can use these credentials to authenticate the following nodes:

- [Kitemaker](#)

Prerequisites

Create a [Kitemaker](#) account.

Supported authentication methods

- API access token

Related resources

Refer to [Kitemaker's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need:

- A **Personal Access Token**: Generate a personal access token from **Manage > Developer settings**. Refer to [API Authentication](#) for more detailed instructions.
-

KoboToolbox credentials

You can use these credentials to authenticate the following nodes:

- [KoboToolbox trigger](#)
- [KoboToolbox](#)

Prerequisites

Create a [KoboToolbox](#) account.

Supported authentication methods

- API token

Related resources

Refer to [KoboToolbox's API documentation](#) for more information about the service.

Using API token

To configure this credential, you'll need:

- An **API Root URL**: Enter the URL of the KoboToolbox server where you created your account. For the Global KoboToolbox Server, use <https://kf.kobotoolbox.org>. For the European Union KoboToolbox Server, use <https://eu.kobotoolbox.org>.
 - An **API Token**: Displayed in your **Account Settings**. Refer to [Getting your API token](#) for more information.
-

LDAP credentials

You can use these credentials to authenticate the following nodes:

- [LDAP](#)

Prerequisites

Create a server directory using Lightweight Directory Access Protocol (LDAP).

Some common LDAP providers include:

- [Jumpcloud](#)
- [Azure ADDS](#)
- [Okta](#)

Supported authentication methods

- LDAP server details

Related resources

Refer to your LDAP provider's own documentation for detailed information.

For general LDAP information, refer to [Basic LDAP concepts](#) for a basic overview and [The LDAP Bind Operation](#) for information on how the bind operation and authentication work.

Using LDAP server details

To configure this credential, you'll need:

- The **LDAP Server Address**: Use the IP address or domain of your LDAP server.
 - The **LDAP Server Port**: Use the number of the port used to connect to the LDAP server.
 - The **Binding DN**: Use the Binding Distinguished Name (Bind DN) for your LDAP server. This is the user account the credential should log in as. If you're using Active Directory, this may look something like `cn=administrator, cn=Users, dc=n8n, dc=io`. Refer to your LDAP provider's documentation for more information on identifying this DN and the related password.
 - The **Binding Password**: Use the password for the **Binding DN** user.
 - Select the **Connection Security**: Options include:
 - None
 - TLS
 - STARTTLS
 - *Optional:* Enter a numeric value in seconds to set a **Connection Timeout**.
-

Lemlist credentials

You can use these credentials to authenticate the following nodes:

- [Lemlist](#)
- [Lemlist Trigger](#)

Prerequisites

Create an account on a [Lemlist](#) instance.

Supported authentication methods

- API key

Related resources

Refer to [Lemlist's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Access your API key in **Settings > Integrations**. Refer to the [API Authentication documentation](#) for more information.
-

Line credentials

You can use these credentials to authenticate the following nodes:

- [Line](#)

Supported authentication methods

- Notify OAuth2

Related resources

Refer to [Line Notify's API documentation](#) for more information about the service.

Using Notify OAuth2

To configure this credential, you'll need a [Line](#) account and:

- A **Client ID**
- A **Client Secret**

To generate both, connect Line with [Line Notify](#). Then:

1. Open the Line Notify page to [add a new service](#).
2. Enter a **Service name**. This name displays when someone tries to connect to the service.
3. Enter a **Service description**.
4. Enter a **Service URL**
5. Enter your **Company/Enterprise**.
6. Select your **Country/region**.
7. Enter your name or team name as the **Representative**.
8. Enter a valid **Email address**. Line will verify this email address before the service is fully registered. Use an email address you have ready access to.
9. Copy the **OAuth Redirect URL** from your n8n credential and enter it as the **Callback URL** in Line Notify.
10. Select **Agree and continue** to agree to the terms of service.
11. Verify the information you entered is correct and select **Add**.
12. Check your email and open the Line Notify Registration URL to verify your email address.
13. Once verification is complete, open [My services](#).
14. Select the service you just added.
15. Copy the **Client ID** and enter it in your n8n credential.
16. Select the option to **Display the Client Secret**. Copy the **Client Secret** and enter it in your n8n credential.
17. In n8n, select **Connect my account** and follow the on-screen prompts to finish the credential.

Refer to the Authentication section of [Line Notify's API documentation](#) for more information.

Linear credentials

You can use these credentials to authenticate the following nodes:

- [Linear Trigger](#)
- [Linear](#)

Prerequisites

Create a [Linear](#) account.

Supported authentication methods

- API key
- OAuth2

Related resources

Refer to [Linear's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- A personal **API Key**: Create a dedicated personal API key in your [Settings > Security & access](#). Refer to the [Linear Personal API keys documentation](#) for more information.

Using OAuth2

To configure this credential, you'll need:

- A **Client ID**: Generated when you create a new OAuth2 application.
- A **Client Secret**: Generated when you create a new OAuth2 application.
- Select the **Actor**: The actor defines how the OAuth2 application should create issues, comments and other changes. Options include:
 - **User** (Linear's default): The application creates resources as the authorizing user. Use this option if you want each user to do their own authentication.
 - **Application**: The application creates resources as itself. Use this option if you have only one user (like an admin) authorizing the application.
- To use this credential with the [Linear Trigger](#) node, you must enable the **Include Admin Scope** toggle.

Refer to the [Linear OAuth2 Authentication documentation](#) for more detailed instructions and explanations. Use the n8n **OAuth Redirect URL** as the **Redirect callback URL** in your Linear OAuth2 application.

LingvaNex credentials

You can use these credentials to authenticate the following nodes:

- [LingvaNex](#)

Prerequisites

Create a [LingvaNex](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Lingvanex's Cloud API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Generate an API key from your **Account** page. Refer to [Where can I get the authorization key?](#) for more detailed instructions.
-

LinkedIn credentials

You can use these credentials to authenticate the following nodes:

- [LinkedIn](#)

Prerequisites

- Create a [LinkedIn](#) account.
- Create a LinkedIn [Company Page](#).

Supported authentication methods

- **Community Management OAuth2**: Use this method if you're a new LinkedIn user or creating a new LinkedIn app.
- **OAuth2**: Use this method for older LinkedIn apps and user accounts.

Related Resources

Refer to [LinkedIn's Community Management API documentation](#) for more information about the service.

This credential works with API version 202404.

Using Community Management OAuth2

Use this method if you're a new LinkedIn user or creating a new LinkedIn app.

To configure this credential, you'll need a [LinkedIn account](#), a LinkedIn [Company Page](#), and:

- A **Client ID**: Generated after you create a new developer app.
- A **Client Secret**: Generated after you create a new developer app.

To create a new developer app and set up the credential:

1. Log into LinkedIn and select this link to [create a new developer app](#).
2. Enter an **App name** for your app, like n8n integration.
3. For the **LinkedIn Page**, enter a LinkedIn [Company Page](#) or use the **Create a new LinkedIn Page** link to create one on-the-fly. Refer to [Associate an App with a LinkedIn Page](#) for more information.
4. Add an **App logo**.
5. Check the box to agree to the **Legal agreement**.
6. Select **Create app**.
7. This should open the **Products** tab. Select the products/APIs you want to enable for your app. For the LinkedIn node to work properly, you must include and configure:
 - **Share on LinkedIn**
 - **Sign In with LinkedIn using OpenID Connect**
 - **Advertising API** (if using it as an organization account rather than an individual)
8. Once you've requested access to the products you need, open the **Auth** tab.
9. Copy the **Client ID** and enter it in your n8n credential.
10. Select the icon to **Copy the Primary Client Secret**. Enter this in your n8n credential as the **Client Secret**.

Refer to [Getting Access to LinkedIn APIs](#) for more information on scopes and permissions.

Using OAuth2

Only use this method for older LinkedIn apps and user accounts.

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

All users must select:

- **Organization Support**: If turned on, the credential requests permission to post as an organization using the w_organization_social scope.

- To use this option, you must put your app through LinkedIn’s [Community Management App Review](#) process.
- **Legacy:** If turned on, the credential uses legacy scopes for `r_liteprofile` and `r_emailaddress` instead of the newer `profile` and `email` scopes.

If you’re [self-hosting](#) n8n, you’ll need to configure OAuth2 from scratch by creating a new developer app:

1. Log into LinkedIn and select this link to [create a new developer app](#).
2. Enter an **App name** for your app, like `n8n integration`.
3. For the **LinkedIn Page**, enter a LinkedIn [Company Page](#) or use the **Create a new LinkedIn Page** link to create one on-the-fly. Refer to [Associate an App with a LinkedIn Page](#) for more information.
4. Add an **App logo**.
5. Check the box to agree to the **Legal agreement**.
6. Select **Create app**.
7. This should open the **Products** tab. Select the products/APIs you want to enable for your app. For the LinkedIn node to work properly, you must include:
 - **Share on LinkedIn**
 - **Sign In with LinkedIn using OpenID Connect**
8. Once you’ve requested access to the products you need, open the **Auth** tab.
9. Copy the **Client ID** and enter it in your n8n credential.
10. Select the icon to **Copy the Primary Client Secret**. Enter this in your n8n credential as the **Client Secret**.

Refer to [Getting Access to LinkedIn APIs](#) for more information on scopes and permissions.

LoneScale credentials

You can use these credentials to authenticate the following nodes:

- [LoneScale](#)
- [LoneScale Trigger](#)

Prerequisites

Create a [LoneScale](#) account.

Supported authentication methods

- API key

Related resources

Refer to [LoneScale’s API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Refer to [LoneScale's Generate an API key documentation](#) to generate your key.
-

Magento 2 credentials

You can use these credentials to authenticate the following node:

- [Magento 2](#)

Prerequisites

- Create a [Magento \(Adobe Commerce\)](#) account.
- Set your store to **Allow OAuth Access Tokens to be used as standalone Bearer tokens**.
 - Go to **Admin > Stores > Configuration > Services > OAuth > Consumer Settings**.
 - Set the **Allow OAuth Access Tokens to be used as standalone Bearer tokens** option to **Yes**.
 - You can also enable this setting from the CLI by running the following command:

```
bin/magento config:set
oauth/consumer/enable_integration_as_bearer 1
```

This step is necessary until n8n updates the Magento 2 credentials to use OAuth. Refer to [Integration Tokens](#) for more information.

Supported authentication methods

- API access token

Related resources

Refer to [Magento's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need:

- A **Host**: Enter the address of your Magento store.
- An **Access Token**: Get an access token from the [Admin Panel](#):
 1. Go to **System > Extensions > Integrations**.
 2. Add a new Integration.
 3. Go to the **API** tab and select the Magento resources you'd like

- the n8n integration to access.
4. From the **Integrations** page, **Activate** the new integration.
 5. Select **Allow** to display your access token so you can copy it and enter it in n8n.
-

Mailcheck credentials

You can use these credentials to authenticate the following nodes:

- [Mailcheck](#)

Prerequisites

Create a [Mailcheck](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Mailcheck's API documentation](#) for more information about the service.

Using API Key

To configure this credential, you'll need:

- An **API Key**: Generate an API Key in the API section of your dashboard. Refer to [Mailcheck's How to create an API key documentation](#) for detailed instructions.
-

Mailchimp credentials

You can use these credentials to authenticate the following nodes:

- [Mailchimp](#)
- [Mailchimp Trigger](#)

Prerequisites

Create a [Mailchimp](#) account.

Supported authentication methods

- API key
- OAuth2

Refer to [Selecting an authentication method](#) for guidance on which method to use.

Related resources

Refer to [Mailchimp's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Generate an API key in the [API keys section](#) of your Mailchimp account. Refer to [Mailchimp's Generate your API key documentation](#) for more detailed instructions.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you need to configure OAuth2 from scratch, [register an application](#). Refer to the [Mailchimp OAuth2 documentation](#) for more information.

Selecting an authentication method

Mailchimp suggests using an API key if you're only accessing your own Mailchimp account's data:

Use an API key if you're writing code that tightly couples *your* application's data to *your* Mailchimp account's data. If you ever need to access *someone else's* Mailchimp account's data, you should be using OAuth 2 ([source](#))

MailerLite credentials

You can use these credentials to authenticate the following nodes:

- [MailerLite](#)
- [MailerLite Trigger](#)

Prerequisites

Create a [MailerLite](#) account.

Supported authentication methods

- API key

Related resources

Refer to [MailerLite's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Generate an API key from the **Integrations** menu. Refer to the [API Authentication documentation](#) for more detailed instructions.

Enable the **Classic API** toggle if the API key is for a MailerLite Classic account instead of the newer MailerLite experience.

Mailgun credentials

You can use these credentials to authenticate the following nodes:

- [Mailgun](#)

Prerequisites

- Create a [Mailgun](#) account.
- [Add and verify a domain](#) in Mailgun or use the provided sandbox domain for testing.

Supported authentication methods

- API key

Related resources

Refer to [Mailgun's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Domain**: If your Mailgun account is based in Europe, select **api.eu.mailgun.net**; otherwise, select **api.mailgun.net**. Refer to [Mailgun Base URLs](#) for more information.
- An **Email Domain**: Enter the email sending domain you're working with. If you have multiple sending domains, refer to [Working with multiple email domains](#) for more information.
- An **API Key**: View your API key in **Settings > API Keys**. Refer to [Mailgun's API Authentication documentation](#) for more detailed instructions.

Working with multiple email domains

If your Mailgun account includes multiple sending domains, create a separate credential for each email domain you're working with.

Mailjet credentials

You can use these credentials to authenticate the following nodes:

- [Mailjet](#)
- [Mailjet Trigger](#)

Prerequisites

Create a [Mailjet](#) account.

Supported authentication methods

- Email API key: For use with Mailjet's Email API
- SMS token: For use with Mailjet's SMS API

Related resources

Refer to [Mailjet's Email API documentation](#) and [Mailjet's SMS API documentation](#) for more information about each service.

Using Email API key

To configure this credential, you'll need:

- An **API Key**: View and generate API keys in your Mailjet [API Key Management](#) page.
- A **Secret Key**: View your API Secret Keys in your Mailjet [API Key Management](#) page.
- *Optional:* Select whether to use **Sandbox Mode** for calls made using this credential. When turned on, all API calls use Sandbox mode: the API will still validate the payloads but won't deliver the actual messages. This can be useful to troubleshoot any payload error messages without actually sending messages. Refer to [Mailjet's Sandbox Mode documentation](#) for more information.

For this credential, you can use either:

- Mailjet's primary API key and secret key
- A subaccount API key and secret key

Refer to Mailjet's [How to create a subaccount \(or additional API key\) documentation](#) for detailed instructions on creating more API keys. Refer to [What are subaccounts and how does it help me?](#) page for more information on Mailjet subaccounts and when you might want to use one.

Using SMS Token

To configure this credential, you'll need:

- An access **Token**: Generate a new token from Mailjet's [SMS Dashboard](#).
-

Malcore credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create a [Malcore](#) account.

Related resources

Refer to [Malcore’s API documentation](#) for more information about authenticating with the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n’s website.

Using API key

To configure this credential, you'll need:

- An **API Key**: Get an API Key from your [Account > API](#).

Refer to [Using the Malcore API](#) for more information.

Mandrill credentials

You can use these credentials to authenticate the following nodes:

- [Mandrill](#)

Prerequisites

- Create a Mailchimp [Transactional email account](#)
- Log in to [Mandrill](#) with your Mailchimp account.

If you already have a Mailchimp account with a Standard plan or higher, enable [Transactional Emails](#) within that account to use Mandrill.

Supported authentication methods

- API key

Related resources

Refer to [Mailchimp's Transactional API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Generate an API key from the Mandrill [Settings](#). Refer to Mailchimp's [Generate your API key documentation](#) for more detailed instructions.
-

Marketstack credentials

You can use these credentials to authenticate the following nodes:

- [Marketstack](#)

Prerequisites

Create a [Marketstack](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Marketstack's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: View and generate API keys in your Marketstack [account dashboard](#).
 - Select whether to **Use HTTPS**: Make this selection based on your Marketstack account plan level:
 - Free plan: Turn off **Use HTTPS**
 - All other plans: Turn on **Use HTTPS**
-

Matrix credentials

You can use these credentials to authenticate the following nodes:

- [Matrix](#)

Prerequisites

Create an account on a [Matrix](#) server. Refer to [Creating an account](#) for more information.

Supported authentication methods

- API access token

Related resources

Refer to the [Matrix Specification](#) for more information about the service.

Refer to the documentation for the specific client you're using to access the Matrix server.

Using API access token

To configure this credential, you'll need:

- An **Access Token**: This token is tied to the account you use to log into Matrix with.
- A **Homeserver URL**: This is the URL of the [homeserver](#) you entered when you created your account. n8n prepopulates this with matrix.org's own server; adjust this if you're using a server hosted elsewhere.

Instructions for getting these details vary depending on the client you're using to access the server. Both the **Access Token** and the **Homeserver URL** can most commonly be found in **Settings > Help & About > Advanced**, but refer to your client's documentation for more details.

Mattermost credentials

You can use these credentials to authenticate the following nodes:

- [Mattermost](#)

Supported authentication methods

- API access token

Related resources

Refer to [Mattermost's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need a [Mattermost](#) account and:

- A personal **Access Token**
- Your Mattermost **Base URL**.

To set it up:

1. In Mattermost, go to **Profile > Security > Personal Access Tokens**.
2. Select **Create Token**.
3. Enter a **Token description**, like n8n integration.
4. Select **Save**.
5. Copy the **Token ID** and enter it as the **Access Token** in your n8n credential.
6. Enter your Mattermost URL as the **Base URL**.
7. By default, n8n connects only if SSL certificate validation succeeds. To connect even if SSL certificate validation fails, turn on **Ignore SSL Issues**.

Refer to the Mattermost [Personal access tokens documentation](#) for more information.

Enable personal access tokens

Not seeing the **Personal Access Tokens** option has two possible causes:

- Mattermost doesn't have the personal access tokens integration enabled.
- You're trying to generate a personal access token as a non-admin user who doesn't have permission to generate personal access tokens.

To identify the root cause and resolve it:

1. Log in to Mattermost as an admin.
2. Go to **System Console > Integrations > Integration Management**.
3. Confirm that **Enable personal access tokens** is set to **true**. If it's not, change.
4. Go to **System Console > User Management > Users**.
5. Search for the user account you want to allow to generate personal access tokens.
6. Select the **Actions** dropdown for the user and select **Manage roles**.
7. Check the box for **Allow this account to generate personal access tokens** and **Save**.

Refer to the Mattermost [Personal access tokens documentation](#) for more information.

Mautic credentials

You can use these credentials to authenticate the following nodes:

- [Mautic](#)
- [Mautic Trigger](#)

Supported authentication methods

- Basic auth
- OAuth2

Related resources

Refer to [Mautic's API documentation](#) for more information about the service.

Using basic auth

To configure this credential, you'll need an account on a [Mautic](#) instance and:

- Your **URL**
- A **Username**
- A **Password**

To set it up:

1. In Mautic, go to **Configuration > API Settings**.
2. If **Enable HTTP basic auth?** is set to **No**, change it to **Yes** and save. Refer to the [API Settings documentation](#) for more information.
3. In n8n, enter the Base **URL** of your Mautic instance.
4. Enter your Mautic **Username**.
5. Enter your Mautic **Password**.

Using OAuth2

To configure this credential, you'll need an account on a [Mautic](#) instance and:

- A **Client ID**: Generated when you create new API credentials.
- A **Client Secret**: Generated when you create new API credentials.
- Your **URL**

To set it up:

1. In Mautic, go to **Configuration > Settings**.
2. Select **API Credentials**.

3. Select the option to **Create new client**.
4. Select **OAuth 2** as the **Authorization Protocol**.
5. Enter a **Name** for your credential, like n8n integration.
6. In n8n, copy the **OAuth Callback URL** and enter it as the **Redirect URI** in Mautic.
7. Select **Apply**.
8. Copy the **Client ID** from Mautic and enter it in your n8n credential.
9. Copy the **Client Secret** from Mautic and enter it in your n8n credential.
10. Enter the Base **URL** of your Mautic instance.

Refer to [What is Mautic's API?](#) for more information.

Enable the API

To enable the API in your Mautic instance:

1. Go to **Settings > Configuration**.
2. Select **API Settings**.
3. Set **API enabled?** to **Yes**.
4. **Save** your changes.

Refer to [How to use the Mautic API](#) for more information.

Medium credentials

You can use these credentials to authenticate the following nodes:

- [Medium](#)

Prerequisites

- Create an account on [Medium](#).
- For OAuth2, request access to credentials by emailing yourfriends@medium.com.

Supported authentication methods

- API access token
- OAuth2

Related resources

Refer to [Medium's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need:

- An **API Access Token**: Generate a token in **Settings > Security and apps > Integration tokens**. Use the integration token this generates as your n8n **Access Token**.

Refer to the Medium API [Self-issued access tokens documentation](#) for more information.

Using OAuth2

To configure this credential, you'll need:

- A **Client ID**
- A **Client Secret**

To generate a **Client ID** and **Client Secret**, you'll need access to the **Developers** menu. From there, create a new application to generate the Client ID and Secret.

Use these settings for your new application:

- Select **OAuth 2** as the **Authorization Protocol**
 - Copy the **OAuth Callback URL** from n8n and use this as the **Callback URL** in Medium.
-

MessageBird credentials

You can use these credentials to authenticate the following nodes:

- [MessageBird](#)

Prerequisites

Create a [Bird](#) account.

Supported authentication methods

- API key

Related resources

Refer to [MessageBird's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: To generate an appropriate key, visit the [Access keys](#) page in MessageBird. Refer to the [API authorization documentation](#) for detailed instructions.
-

Metabase credentials

You can use these credentials to authenticate the following nodes:

- [Metabase node](#)

Prerequisites

Create a [Metabase](#) account with access to a Metabase instance.

Supported authentication methods

- Basic auth

Related resources

Refer to [Metabase's API documentation](#) for more information about the service.

Using basic auth

To configure this credential, you'll need:

- A **URL**: Enter the base URL of your Metabase instance. If you're using a custom domain, use that URL.
 - A **Username**: Enter your Metabase username.
 - A **Password**: Enter your Metabase password.
-

Microsoft credentials

You can use these credentials to authenticate the following nodes:

- [Microsoft Dynamics CRM](#)
- [Microsoft Excel](#)
- [Microsoft Graph Security](#)
- [Microsoft OneDrive](#)
- [Microsoft Outlook](#)
- [Microsoft SharePoint](#)
- [Microsoft Teams](#)
- [Microsoft Teams Trigger](#)
- [Microsoft To Do](#)

Prerequisites

- Create a [Microsoft Azure](#) account.
- Create at least one user account with access to the appropriate service.
- If a corporate Microsoft Entra account manages the user account, the administrator account has enabled the option “User can consent to apps accessing company data on their behalf” for this user (see the [Microsoft Entra documentation](#)).

Supported authentication methods

- OAuth2

Related resources

Refer to the linked Microsoft API documentation below for more information about each service’s API:

- Dynamics CRM: [Web API](#)
- Excel: [Graph API](#)
- Graph Security: [Graph API](#)
- OneDrive: [Graph API](#)
- Outlook: [Graph API](#) and [Outlook API](#)
- Teams: [Graph API](#)
- To Do: [Graph API](#)

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

Some Microsoft services require extra information for OAuth2. Refer to [Service-specific settings](#) for more guidance on those services.

For self-hosted users, there are two main steps to configure OAuth2 from scratch:

1. [Register an application](#) with the Microsoft Identity Platform.
2. [Generate a client secret](#) for that application.

Follow the detailed instructions for each step below. For more detail on the Microsoft OAuth2 web flow, refer to [Microsoft authentication and authorization basics](#).

Register an application

Register an application with the Microsoft Identity Platform:

1. Open the [Microsoft Application Registration Portal](#).
2. Select **Register an application**.
3. Enter a **Name** for your app.
4. In **Supported account types**, select **Accounts in any organizational directory (Any Azure AD directory - Multi-tenant) and personal Microsoft accounts (for example, Skype, Xbox)**.
5. In **Register an application**:
 1. Copy the **OAuth Callback URL** from your n8n credential.

2. Paste it into the **Redirect URI (optional)** field.
3. Select **Select a platform > Web**.
6. Select **Register** to finish creating your application.
7. Copy the **Application (client) ID** and paste it into n8n as the **Client ID**.

Refer to [Register an application with the Microsoft Identity Platform](#) for more information.

Generate a client secret

With your application created, generate a client secret for it:

1. On your Microsoft application page, select **Certificates & secrets** in the left navigation.
2. In **Client secrets**, select **+ New client secret**.
3. Enter a **Description** for your client secret, such as `n8n credential`.
4. Select **Add**.
5. Copy the **Secret** in the **Value** column.
6. Paste it into n8n as the **Client Secret**.
7. If you see other fields in the `n8n credential`, refer to [Service-specific settings](#) below for guidance on completing those fields.
8. Select **Connect my account** in n8n to finish setting up the connection.
9. Log in to your Microsoft account and allow the app to access your info.

Refer to Microsoft's [Add credentials](#) for more information on adding a client secret.

Service-specific settings

The following services require extra information for OAuth2:

Dynamics

Dynamics OAuth2 requires information about your Dynamics domain and region. Follow these extra steps to complete the credential:

1. Enter your Dynamics **Domain**.
2. Select the Dynamics data center **Region** you're within.

Refer to the [Microsoft Datacenter regions documentation](#) for more information on the region options and corresponding URLs.

Microsoft (general)

The general Microsoft OAuth2 also requires you to provide a space-separated list of **Scopes** for this credential.

Refer to [Scopes and permissions in the Microsoft identity platform](#) for a list of possible scopes.

Outlook

Outlook OAuth2 supports the credential accessing a user's primary email inbox or a shared inbox. By default, the credential will access a user's primary email inbox. To change this behavior:

1. Turn on **Use Shared Inbox**.
2. Enter the target user's UPN or ID as the **User Principal Name**.

SharePoint

SharePoint OAuth2 requires information about your SharePoint **Subdomain**.

To complete the credential, enter the **Subdomain** part of your SharePoint URL. For example, if your SharePoint URL is <https://tenant123.sharepoint.com>, the subdomain is tenant123.

SharePoint requires the following permissions:

Application permissions:

- `Sites.Read.All`
- `Sites.ReadWrite.All`

Delegated permissions:

- `SearchConfiguration.Read.All`
- `SearchConfiguration.ReadWrite.All`

Common issues

Here are the known common errors and issues with Microsoft OAuth2 credentials.

-8<- “_snippets/integrations/builtin/credentials/microsoft-need-admin-approval.md”

Microsoft Azure Monitor credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

- Create a Microsoft Azure account or subscription
- An app registered in Microsoft Entra ID

Supported authentication methods

- OAuth2

Related resources

Refer to [Microsoft Azure Monitor's API documentation](#) for more information about the service.

Using OAuth2

To configure this credential, you'll need a Microsoft Azure account and:

- A **Client ID**
- A **Client Secret**
- A **Tenant ID**
- The **Resource** you plan to access

Refer to [Microsoft Azure Monitor's API documentation](#) for more information about authenticating to the service.

Microsoft Entra ID credentials

You can use these credentials to authenticate the following nodes:

- [Microsoft Entra ID](#)

Prerequisites

- Create a Microsoft Entra ID account or subscription.
- If the user account is managed by a corporate Microsoft Entra account, the administrator account has enabled the option “User can consent to apps accessing company data on their behalf” for this user (see the [Microsoft Entra documentation](#)).

Microsoft includes an Entra ID free plan when you create a [Microsoft Azure](#) account.

Supported authentication methods

- OAuth2

Related resources

Refer to [Microsoft Entra ID's documentation](#) for more information about the service.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

For self-hosted users, there are two main steps to configure OAuth2 from scratch:

1. [Register an application](#) with the Microsoft Identity Platform.
2. [Generate a client secret](#) for that application.

Follow the detailed instructions for each step below. For more detail on the Microsoft OAuth2 web flow, refer to [Microsoft authentication and authorization basics](#).

Register an application

Register an application with the Microsoft Identity Platform:

1. Open the [Microsoft Application Registration Portal](#).
2. Select **Register an application**.
3. Enter a **Name** for your app.
4. In **Supported account types**, select **Accounts in any organizational directory (Any Azure AD directory - Multi-tenant) and personal Microsoft accounts (for example, Skype, Xbox)**.
5. In **Register an application**:
 1. Copy the **OAuth Callback URL** from your n8n credential.
 2. Paste it into the **Redirect URI (optional)** field.
 3. Select **Select a platform > Web**.
6. Select **Register** to finish creating your application.
7. Copy the **Application (client) ID** and paste it into n8n as the **Client ID**.

Refer to [Register an application with the Microsoft Identity Platform](#) for more information.

Generate a client secret

With your application created, generate a client secret for it:

1. On your Microsoft application page, select **Certificates & secrets** in the left navigation.
2. In **Client secrets**, select **+ New client secret**.
3. Enter a **Description** for your client secret, such as n8n credential.
4. Select **Add**.
5. Copy the **Secret** in the **Value** column.
6. Paste it into n8n as the **Client Secret**.
7. Select **Connect my account** in n8n to finish setting up the connection.
8. Log in to your Microsoft account and allow the app to access your info.

Refer to Microsoft's [Add credentials](#) for more information on adding a client secret.

Setting custom scopes

Microsoft Entra ID credentials use the following scopes by default:

- [`openid`](#)
- [`offline_access`](#)
- [`AccessReview.ReadWrite.All`](#)
- [`Directory.ReadWrite.All`](#)
- [`NetworkAccessPolicy.ReadWrite.All`](#)
- [`DelegatedAdminRelationship.ReadWrite.All`](#)
- [`EntitlementManagement.ReadWrite.All`](#)
- [`User.ReadWrite.All`](#)
- [`Directory.AccessAsUser.All`](#)

- [Sites.FullControl.All](#)
- [GroupMember.ReadWrite.All](#)

To select different scopes for your credentials, enable the **Custom Scopes** slider and edit the **Enabled Scopes** list. Keep in mind that some features may not work as expected with more restrictive scopes.

Common issues

Here are the known common errors and issues with Microsoft Entra credentials.

-8<- “_snippets/integrations/builtin/credentials/microsoft-need-admin-approval.md”

Microsoft SQL credentials

You can use these credentials to authenticate the following nodes:

- [Microsoft SQL](#)

Prerequisites

Create a user account on a [Microsoft SQL server](#) database.

Supported authentication methods

- SQL database connection

Related resources

Refer to [Microsoft’s Connect to SQL Server documentation](#) for more information about connecting to the service.

Using SQL database connection

To configure this credential, you’ll need:

- The **Server** name
- The **Database** name
- Your **User** account/**ID**
- Your **Password**
- The **Port** to use for the connection
- The **Domain** name
- Whether to use **TLS**
- Whether to **Ignore SSL Issues**
- The **Connect Timeout**
- The **Request Timeout**
- The **TDS Version** the connection should use

To set up the database connection:

1. Enter the SQL Server Host Name as the **Server**. In an existing SQL Server connection, the host name comes before the instance name in the format HOSTNAME\INSTANCENAME. Find the host name:
 - In the **Object Explorer** pane as the top-level object for your database.
 - In the footer of a query window.
 - Viewing the current connection **Properties** and looking for **Name** or **Display Name**.
 - Refer to [Find SQL Server Instance Name | When you're connected to SQL Server](#) for more information. You can also find the information in the [Error logs](#).
2. Enter the SQL Server Instance Name as the **Database** name. Find this name using the same steps listed above for finding the host name.
 - If you don't see an instance name in any of these places, then your database uses the default MSSQLSERVER instance name.
3. Enter your **User** account name or ID.
4. Enter your **Password**.
5. For the **Port**:
 - SQL Server defaults to 1433.
 - If you can't connect over port 1433, check the [Error logs](#) for the phrase `Server is listening on` to identify the port number you should enter.
6. You only need to enter the **Domain** name if users in multiple domains access your database. Run this SQL query to get the domain name:


```
SELECT DEFAULT_DOMAIN() [DomainName];
```
7. Select whether to use **TLS**.
8. Select whether to **Ignore SSL Issues**: If turned on, the credential will connect even if SSL certificate validation fails.
9. Enter the number of milliseconds n8n should attempt the initial connection to complete before disconnecting as the **Connect Timeout**. Refer to the [SqlConnection.ConnectionTimeout property documentation](#) for more information.
 - SQL Server stores this timeout as seconds, while n8n stores it as milliseconds. If you're copying your SQL Server defaults, multiply by 100 before entering the number here.
10. Enter the number of milliseconds n8n should wait on a given request before timing out as the **Request Timeout**. This is basically a query timeout parameter. Refer to [Troubleshoot query time-out errors](#) for more information.
11. Select the Tabular Data Stream (TDS) protocol to use from the **TDS Version** dropdown. If the server doesn't support the version you select here, the connection uses a negotiated alternate version. Refer to [Appendix A: Product Behavior](#) for a more detailed breakdown of the TDS versions' compatibility with different SQL Server versions and .NET frameworks. Options include:

- **7_4 (SQL Server 2012 ~ 2019)**: TDS version 7.4.
 - **7_3_B (SQL Server 2008R2)**: TDS version 7.3.B.
 - **7_3_A (SQL Server 2008)**: TDS version 7.3.A.
 - **7_2 (SQL Server 2005)**: TDS version 7.2.
 - **7_1 (SQL Server 2000)**: TDS version 7.1.
-

Milvus credentials

You can use these credentials to authenticate the following nodes:

- [Milvus Vector Store](#)

Prerequisites

Create and run an [Milvus](#) instance. Refer to the [Install Milvus](#) for more information.

Supported authentication methods

- Basic auth

Related resources

Refer to [Milvus's Authentication documentation](#) for more information about setting up authentication.

-8<- “[snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md](#)”

Using basic auth

To configure this credential, you'll need:

- **Base URL**: The base URL of your Milvus instance. The default is `http://localhost:19530`.
 - **Username**: The username to authenticate to your Milvus instance. The default value is `root`.
 - **Password**: The password to authenticate to your Milvus instance. The default value is `Milvus`.
-

Mindee credentials

You can use these credentials to authenticate the following nodes:

- [Mindee](#)

Prerequisites

Create a [Mindee](#) account.

Supported authentication methods

- Invoice API key: For use with the [Invoice OCR API](#)
- Receipt API key: For use with the [Receipt OCR API](#)

Related resources

Refer to [Mindee's Invoice OCR API documentation](#) and [Mindee's Receipt OCR API documentation](#) for more information about each service.

Using invoice API key

To configure this credential, you'll need:

- An **API Key**: Refer to the Mindee [Create & Manage API Keys documentation](#) for instructions on creating API keys.

Using receipt API key

To configure this credential, you'll need:

- An **API Key**: Refer to the Mindee [Create & Manage API Keys documentation](#) for instructions on creating API keys.
-

Miro credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create a [Miro](#) account.

Supported authentication methods

- OAuth2

Related resources

Refer to [Miro's API documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows](#) and [related content](#) on n8n's website.

Using OAuth2

To configure this credential, you'll need a [Miro](#) account and app, as well as:

- A **Client ID**: Generated when you create a new OAuth2 application.
- A **Client Secret**: Generated when you create a new OAuth2 application.

Refer to [Miro's API documentation](#) for more information about authenticating to the service.

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you're [self-hosting](#) n8n, you'll need to [create an app](#) to configure OAuth2. Refer to [Miro's OAuth documentation](#) for more information about setting up OAuth2.

MISP credentials

You can use these credentials to authenticate the following nodes:

- [MISP](#)

Prerequisites

Install and run a [MISP](#) instance.

Supported authentication methods

- API key

Related resources

Refer to [MISP's Automation API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: In MISP, these are called Automation keys. Get an automation key from **Event Actions > Automation**. Refer to [MISP's automation keys documentation](#) for instructions on generating more keys.
 - A **Base URL**: Your MISP URL.
 - Select whether to **Allow Unauthorized Certificates**: If turned on, the credential will connect even if SSL certificate validation fails.
-

Mist credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create a [Mist](#) account and organization. Refer to [Create a Mist account and Organization](#) for detailed instructions.

Supported authentication methods

- API token

Related resources

Refer to [Mist’s documentation](#) for more information about the service. If you’re logged in to your Mist account, go to <https://api.mist.com/api/v1/docs/Home> to view the full API documentation.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n’s website.

Using API token

To configure this credential, you’ll need:

- An **API Token**: You can use either a User API token or an Org API token. Refer to [How to generate a user API token](#) for instructions on generating a User API token. Refer to [Org API token](#) for instructions on generating an Org API token.
- Select the **Region** you’re in. Options include:
 - **Europe**: Select this option if your cloud environment is in any of the EMEA regions.
 - **Global**: Select this option if your cloud environment is in any of the global regions.

Mistral Cloud credentials

You can use these credentials to authenticate the following nodes:

- [Mistral AI](#)
- [Mistral Cloud](#)
- [Embeddings Mistral Cloud](#)

Prerequisites

- Create a [Mistral](#) La Plateforme account.
- You must add payment information in **Workspace > Billing** and

activate payments to enable API keys. Refer to [Account setup](#) for more information.

Supported authentication methods

- API key

Related resources

Refer to [Mistral's API documentation](#) for more information about the APIs.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Using API key

To configure this credential, you'll need:

- An **API Key**

Once you've added payment information to your Mistral Cloud account:

1. Sign in to your [Mistral account](#).
2. Go to the **API Keys** page.
3. Select **Create new key**.
4. Copy the API key and enter it in your n8n credential.

Refer to [Account setup](#) for more information.

// note | Paid account required
Mistral requires you to add payment information and activate payments to use API keys. Refer to the [Prerequisites](#) section above for more information.

Mocean credentials

You can use these credentials to authenticate the following nodes:

- [Mocean](#)

Prerequisites

Create a [Mocean](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Mocean's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**
- An **API Secret**

Both the key and secret are accessible in your Mocean [Dashboard](#). Refer to [API Authentication](#) for more information.

monday.com credentials

You can use these credentials to authenticate the following nodes:

- [monday.com](#)

Supported authentication methods

- API token
- OAuth2

Related resources

Refer to [monday.com's API documentation](#) for more information about authenticating with the service.

Using API token

To configure this credential, you'll need a [monday.com](#) account and:

- An **API Token V2**

To get your token:

1. In your monday.com account, select your profile picture in the top right corner.
2. Select **Developers**. The Developer Center opens in a new tab.
3. In the Developer Center, select **My Access Tokens > Show**.
4. Copy your personal token and enter it in your n8n credential as the **Token V2**.

Refer to [monday.com API Authentication](#) for more information.

Using OAuth2

To configure this credential, you'll need a [monday.com](#) account and:

- A **Client ID**
- A **Client Secret**

To generate both these fields, register a new monday.com application:

1. In your monday.com account, select your profile picture in the top right corner.
2. Select **Developers**. The Developer Center opens in a new tab.
3. In the Developer Center, select **Build app**. The app details open.
4. Enter a **Name** for your app, like n8n integration.
5. Copy the **Client ID** and enter it in your n8n credential.
6. **Show the Client Secret**, copy it, and enter it in your n8n credential.
7. In the left menu, select **OAuth**.
8. For **Scopes**, select boards:write and boards:read.
9. Select **Save Scopes**.
10. Select the **Redirect URLs** tab.
11. Copy the **OAuth Redirect URL** from n8n and enter it as the **Redirect URL**.
12. **Save** your changes in monday.com.
13. In n8n, select **Connect my account** to finish the setup.

Refer to [Create an app](#) for more information on creating apps.

Refer to [OAuth and permissions](#) for more information on the available scopes and setting up the Redirect URL.

MongoDB credentials

You can use these credentials to authenticate the following nodes:

- [MongoDB](#)
- [MongoDB Atlas Vector Store](#)
- [MongoDB Chat Memory](#)

Prerequisites

- Create a user account with the appropriate permissions on a [MongoDB](#) server.
- As a Project Owner, add all the [n8n IP addresses](#) to the IP Access List Entries in the project's **Network Access**. Refer to [Add IP Access List entries](#) for detailed instructions.

If you are setting up MongoDB from scratch, create a cluster and a database. Refer to the [MongoDB Atlas documentation](#) for more detailed instructions on these steps.

Supported authentication methods

- Database connection - Connection string
- Database connection - Values

Related resources

Refer to the [MongoDBs Atlas documentation](#) for more information about the service.

Using database connection - Connection string

To configure this credential, you'll need the [Prerequisites](#) listed above. Then:

1. Select **Connection String** as the **Configuration Type**.
2. Enter your MongoDB **Connection String**. To get your connection string in MongoDB, go to **Database > Connect**.
 1. Select **Drivers**.
 2. Copy the code you see in **Add your connection string into your application code**. It will be something like:

```
mongodb+srv://yourName:yourPassword@clusterName.mongodb.net/?retryWrites=true&w=majority.
```
 3. Replace the <password> and <username> in the connection string with the database user's credentials you'll be using.
 4. Enter that connection string into n8n.
 5. Refer to [Connection String](#) for information on finding and formatting your connection string.
3. Enter your **Database** name. This is the name of the database that the user whose details you added to the connection string is logging into.
4. Select whether to **Use TLS**: Turn on to use TLS. You must have your MongoDB database configured to use TLS and have an x.509 certificate generated. Add information for these certificate fields in n8n:
 - **CA Certificate**
 - **Public Client Certificate**
 - **Private Client Key**
 - **Passphrase**

Refer to [MongoDB's x.509 documentation](#) for more information on working with x.509 certificates.

Using database connection - Values

To configure this credential, you'll need the [Prerequisites](#) listed above. Then:

1. Select **Values** as the **Configuration Type**.
2. Enter the database **Host** name or address.
3. Enter the **Database** name.
4. Enter the **User** you'd like to log in as.
5. Enter the user's **Password**.
6. Enter the **Port** to connect over. This is the port number your server uses to listen for incoming connections.
7. Select whether to **Use TLS**: Turn on to use TLS. You must have your MongoDB database configured to use TLS and have an x.509 certificate generated. Add information for these certificate fields in n8n:
 - **CA Certificate**
 - **Public Client Certificate**
 - **Private Client Key**
 - **Passphrase**

Refer to [MongoDB's x.509 documentation](#) for more information on working with x.509 certificates.

Monica CRM credentials

You can use these credentials to authenticate the following nodes:

- [Monica CRM](#)

Prerequisites

Sign up for a [Monica CRM](#) account or self-host an instance.

Supported authentication methods

- API token

Related resources

Refer to [Monica's API documentation](#) for more information about the service.

Using API token

To configure this credential, you'll need:

- Your **Environment**:
 - Select **Cloud-Hosted** if you access your Monica instance through Monica.
 - Select **Self-Hosted** if you have self-hosted Monica on your own server. Provide your **Self-Hosted Domain**.
 - An **API Token**: Generate a token in **Settings > API**.
-

Motorhead credentials

You can use these credentials to authenticate the following nodes:

- [Motorhead](#)

Supported authentication methods

- API key

Related resources

Refer to [Motorhead's API documentation](#) for more information about the service.

-8<- “ snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Using API key

To configure this credential, you'll need a [Motorhead](#) account and:

- Your **Host** URL
- An **API Key**
- A **Client ID**

To set it up, you'll generate an API key:

1. If you're self-hosting Motorhead, update the **Host** URL to match your Motorhead URL.
2. In Motorhead, go to **Settings > Organization**.
3. In the **API Keys** section, select **Create**.
4. Enter a **Name** for your API Key, like n8n integration.
5. Select **Generate**.
6. Copy the **apiKey** and enter it in your n8n credential.
7. Return to the API key list.
8. Copy the **clientId** for the key and enter it as the **Client ID** in your n8n credential.

Refer to [Generate an API key](#) for more information.

MQTT credentials

You can use these credentials to authenticate the following nodes:

- [MQTT](#)
- [MQTT Trigger](#)

Prerequisites

Install an [MQTT broker](#).

MQTT provides a list of Servers/Brokers at [MQTT Software](#).

Supported authentication methods

- Broker connection

Related resources

Refer to [MQTT's documentation](#) for more information about the MQTT protocol.

Refer to your broker provider's documentation for more detailed configuration and details.

Using broker connection

To configure this credential, you'll need:

- Your MQTT broker's **Protocol**
- The **Host**
- The **Port**
- A **Username** and **Password** to authenticate with
- If you're using **SSL**, the relevant certificates and keys

To set things up:

1. Select the broker's **Protocol**, which determines the URL n8n uses. Options include:
 - **Mqtt**: Begin the URL with the standard mqtt: protocol.
 - **Mqtts**: Begin the URL with the secure mqtts: protocol.
 - **Ws**: Begin the URL with the WebSocket ws: protocol.
2. Enter your broker **Host**.
3. Enter the **Port** number n8n should use to connect to the broker host.
4. Enter the **Username** to log into the broker as.
5. Enter that user's **Password**.
6. If you want to receive QoS 1 and 2 messages while offline, turn off the **Clean Session** toggle.
7. Enter a **Client ID** you'd like the credential to use. If you leave this blank, n8n will generate one for you. You can use a fixed or expression-based Client ID.
 - Client IDs can be useful to identify and track connection access. n8n recommends using something with n8n in it for easier auditing.
8. If your MQTT broker uses SSL, turn the **SSL** toggle on. Once you turn it on:
 1. Select whether to use **Passwordless** connection with certificates, which is like the SASL mechanism EXTERNAL. If turned on:
 1. Select whether to **Reject Unauthorized Certificate**: If turned off, n8n will connect even if the certificate validation fails.
 2. Add an SSL **Client Certificate**.
 3. Add an SSL **Client Key** for the Client Certificate.
 2. One or more SSL **CA Certificates**.

Refer to your MQTT broker provider's documentation for more detailed configuration instructions.

MSG91 credentials

You can use these credentials to authenticate the following nodes:

- [MSG91](#)

Prerequisites

Create a [MSG91](#) account.

Supported authentication methods

- API key

Related resources

Refer to [MSG91's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **Authentication Key**: To get your Authentication Key, go to the user menu and select **Authkey**. Refer to MSG91's [Where can I find my authentication key? documentation](#) for more information.

IP Security

MSG91 enables IP Security by default for authkeys.

For the n8n credentials to function with this setting enabled, add all the [n8n IP addresses](#) as whitelisted IPs in MSG91. You can add them in one of two places, depending on your desired security level:

- To allow any/all authkeys in the account to work with n8n, add the n8n IP addresses in the **Company's whitelisted IPs** section of the **Authkey** page.
 - To allow only specific authkeys to work with n8n, add the n8n IP addresses in the **Whitelisted IPs** section of an authkey's details.
-

MySQL credentials

You can use these credentials to authenticate the following nodes:

- [MySQL](#)
- [Agent](#)

Prerequisites

Create a user account on a [MySQL](#) server database.

Supported authentication methods

- Database connection

Related resources

Refer to [MySQL's documentation](#) for more information about the service.

Using database connection

To configure this credential, you'll need:

- The server **Host**: The database's host name or IP address.
- The **Database** name.
- A **User** name.
- A **Password** for that user.
- The **Port** number used by the MySQL server.
- **Connect Timeout**: The number of milliseconds during the initial database connection before a timeout occurs.
- **SSL**: If your database is using SSL, turn this on and add details for the SSL certificate.
- **SSH Tunnel**: Choose whether to connect over an SSH tunnel. An SSH tunnel lets un-encrypted traffic pass over an encrypted connection and enables authorized remote access to servers protected from outside connections by a firewall.

To set up your database connection credential:

1. Enter your database's hostname as the **Host** in your n8n credential. Run this query to confirm the hostname:

```
SHOW VARIABLES WHERE Variable_name = 'hostname';
```

2. Enter your database's name as the **Database** in your n8n credential. Run this query to confirm the database name:

```
SHOW DATABASES;
```

3. Enter the username of a **User** in the database. This user should have appropriate permissions for whatever actions you want n8n to perform.

4. Enter the **Password** for that user.

5. Enter the **Port** number used by the MySQL server (default is 3306). Run this query to confirm the port number:

```
SHOW VARIABLES WHERE Variable_name = 'port';
```

6. Enter the **Connect Timeout** you'd like the node to use. The Connect Timeout is the number of milliseconds during the initial database connection the node should wait before timing out. n8n defaults to 10000 which is the default used by MySQL of 10 seconds. If you want to match your database's `connect_timeout`, run this query to get it, then multiply by 1000 before entering it in n8n:

```
SHOW VARIABLES WHERE Variable_name = 'connect_timeout';
```

7. If your database uses SSL and you'd like to use **SSL** for the connection, turn this option on in the credential. If you turn it on, enter the information from your MySQL SSL certificate in these fields:

1. Enter the `ca.pem` file contents in the **CA Certificate** field.
2. Enter the `client-key.pem` file contents in the **Client Private Key** field.

3. Enter the `client-cert.pem` file contents in the **Client Certificate** field.
8. If you want to use **SSH Tunnel** for the connection, turn this option on in the credential. Otherwise, skip it. If you turn it on:

1. Select the **SSH Authenticate with** to set the SSH Tunnel type to build:
 - Select **Password** if you want to connect to SSH using a password.
 - Select **Private Key** if you want to connect to SSH using an identity file (private key) and a passphrase.
2. Enter the **SSH Host**. n8n uses this host to create the SSH URI formatted as: [user@]host:port.
3. Enter the **SSH Port**. n8n uses this port to create the SSH URI formatted as: [user@]host:port.
4. Enter the **SSH User** to connect with. n8n uses this user to create the SSH URI formatted as: [user@]host:port.
5. If you selected **Password** for **SSH Authenticate with**, add the **SSH Password**.
6. If you selected **Private Key** for **SSH Authenticate with**:
 1. Add the contents of the **Private Key** or identity file used for SSH. This is the same as using the `ssh-identity-file` option with the `shell-connect()` command in MySQL.
 2. If the **Private Key** was created with a passphrase, enter that **Passphrase**. This is the same as using the `ssh-identity-pass` option with the `shell-connect()` command in MySQL. If the **Private Key** has no passphrase, leave this field blank.

Refer to [MySQL | Creating SSL and RSA Certificates and Keys](#) for more information on working with SSL certificates in MySQL. Refer to [MySQL | Using an SSH Tunnel](#) for more information on working with SSH tunnels in MySQL.

NASA credentials

You can use these credentials to authenticate the following nodes:

- [NASA](#)

Supported authentication methods

- API key

Related resources

Refer to the **Browse APIs** section of the [NASA Open APIs](#) for more information about the service.

Using an API key

To configure this credential, you'll need:

- An **API Key**

To generate an API key:

1. Go to the [NASA Open APIs](#) page.

2. Complete the fields in the **Generate API Key** section.
 3. Copy the **API Key** and enter it in your n8n credential.
-

Netlify credentials

You can use these credentials to authenticate the following nodes:

- [Netlify](#)
- [Netlify Trigger](#)

Prerequisites

Create a [Netlify](#) account.

Supported authentication methods

- API access token

Related resources

Refer to [Netlify's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need:

- An **Access Token**: Generate an Access Token in **Applications > Personal Access Tokens**. Refer to [Netlify API Authentication](#) for more detailed instructions.
-

Netscaler ADC credentials

You can use these credentials to authenticate the following nodes:

- [Netscaler ADC node](#)

Prerequisites

Install a [NetScaler/Citrix ADC appliance](#).

Supported authentication methods

- Basic auth

Related resources

Refer to [Netscaler ADC's 14.1 NITRO API documentation](#) for more information about the service.

Using basic auth

To configure this credential, you'll need:

- A **URL**: Enter the URL of your NetScaler/Citrix ADC instance.
- A **Username**: Enter your NetScaler/Citrix ADC username.
- A **Password**: Enter your NetScaler/Citrix ADC password.

Refer to [Performing Basic Netscaler ADC Operations](#) for more information.

Nextcloud credentials

You can use these credentials to authenticate the following nodes:

- [Nextcloud](#)

Supported authentication methods

- Basic auth
- OAuth2

Related resources

Refer to [Nextcloud's API documentation](#) for more information about the service.

Refer to [Nextcloud's user manual](#) for more information on installing and configuring Nextcloud.

Using basic auth

To configure this credential, you'll need a [Nextcloud](#) account and:

- Your **WebDAV URL**
- Your **User** name
- Your **Password** or an app password

To set it up:

1. To create your **WebDAV URL**: If Nextcloud is in the root of your domain: Enter the URL you use to access Nextcloud and add /remote.php/webdav/. For example, if you access Nextcloud at <https://cloud.n8n.com>, your WebDAV URL is <https://cloud.n8n.com/remote.php/webdav>.
 - If you have Nextcloud installed in a subdirectory, enter the URL you use to access Nextcloud and add /<subdirectory>/remote.php/webdav/. Replace <subdirectory> with the subdirectory Nextcloud's installed in.
 - Refer to Nextcloud's [Third-party WebDAV clients](#)

documentation for more information on constructing your WebDAV URL.

2. Enter your **User** name.
3. For the **Password**, Nextcloud recommends using an app password rather than your user password. To create an app password:
 1. In the Nextcloud Web interface, select your avatar in the top right and select **Personal settings**.
 2. In the left menu, choose **Security**.
 3. Scroll to the bottom to the **App Password** section and create a new app password.
 4. Copy that app password and enter it in n8n as your **Password**.

Using OAuth2

To configure this credential, you'll need a [Nextcloud](#) account and:

- An **Authorization URL** and **Access Token URL**: These depend on the URL you use to access Nextcloud.
- A **Client ID**: Generated once you add an OAuth2 client application in **Administrator Security Settings**.
- A **Client Secret**: Generated once you add an OAuth2 client application in **Administrator Security Settings**.
- A **WebDAV URL**: This depends on the URL you use to access Nextcloud.

To set it up:

1. In Nextcloud, open your **Administrator Security Settings**.
2. Find the **Add client** section under **OAuth 2.0 clients**.
3. Enter a **Name** for your client, like `n8n integration`.
4. Copy the **OAuth Callback URL** from n8n and enter it as the **Redirection URI**.
5. Then select **Add** in Nextcloud.
6. In n8n, update the **Authorization URL** to replace `https://nextcloud.example.com` with the URL you use to access Nextcloud. For example, if you access Nextcloud at `https://cloud.n8n.com`, the Authorization URL is `https://cloud.n8n.com/apps/oauth2/authorize`.
7. In n8n, update the **Access Token URL** to replace `https://nextcloud.example.com` with the URL you use to access Nextcloud. For example, if you access Nextcloud at `https://cloud.n8n.com`, the Access Token URL is `https://cloud.n8n.com/apps/oauth2/api/v1/token`.
8. Copy the Nextcloud **Client Identifier** for your OAuth2 client and enter it as the **Client ID** in n8n.
9. Copy the Nextcloud **Secret** and enter it as the **Client Secret** in n8n.
10. In n8n, to create your **WebDAV URL**: If Nextcloud is in the root of your domain, enter the URL you use to access Nextcloud and add `/remote.php/webdav/`. For example, if you access Nextcloud at

<https://cloud.n8n.com>, your WebDAV URL is
<https://cloud.n8n.com/remote.php/webdav>.

- If you have Nextcloud installed in a subdirectory, enter the URL you use to access Nextcloud and add /<subdirectory>/remote.php/webdav/. Replace <subdirectory> with the subdirectory Nextcloud's installed in.
- Refer to Nextcloud's [Third-party WebDAV clients](#) documentation for more information on constructing your WebDAV URL.

Refer to the Nextcloud [OAuth2 Configuration documentation](#) for more detailed instructions.

NocoDB credentials

You can use these credentials to authenticate the following nodes:

- [NocoDB](#)

Supported authentication methods

- API token (recommended)
- User auth token

Related resources

Refer to [NocoDB's API documentation](#) for more information about the service.

Using API token

To configure this credential, you'll need a [NocoDB](#) instance and:

- An **API Token**
- Your database **Host**

To generate an API token:

1. Log into NocoDB and select the **User menu** in the bottom left sidebar.
2. Select **Account Settings**.
3. Open the **Tokens** tab.
4. Select **Add new API token**.
5. Enter a **Name** for your token, like `n8n integration`.
6. Select **Save**.
7. Copy the **API Token** and enter it in your n8n credential.
8. Enter the **Host** of your NocoDB instance in your n8n credential, for example `http://localhost:8080`.

Refer to the NocoDB [API Tokens documentation](#) for more detailed instructions.

Using user auth token

Before NocoDB deprecated it, user auth token was a temporary token designed for quick experiments with the API, valid for a session until the user logs out or for 10 hours.

To configure this credential, you'll need a [NocoDB](#) instance and:

- A **User Token**
- Your database **Host**

To generate a user auth token:

1. Log into NocoDB and select the **User menu** in the bottom left sidebar.
2. Select **Copy Auth token**.
3. Enter that auth token as the **User Token** in n8n.
4. Enter the **Host** of your NocoDB instance, for example
`http://localhost:8080`.

Refer to the NocoDB [Auth Tokens documentation](#) for more information.

Notion credentials

You can use these credentials to authenticate the following nodes:

- [Notion](#)
- [Notion Trigger](#)

Prerequisites

Create a [Notion](#) account with admin level access.

Supported authentication methods

- API integration token: Used for internal integrations.
- OAuth2: Used for public integrations.

Related resources

Refer to [Notion's API documentation](#) for more information about the service.

Using API integration token

To configure this credential, you'll need:

- An **Internal Integration Secret**: Generated once you create a Notion integration.

To generate an integration secret, [create a Notion integration](#) and grab the integration secret from the **Secrets** tab:

1. Go to your Notion [integration dashboard](#).
2. Select the **+ New integration** button.
3. Enter a **Name** for your integration, for example n8n integration. If desired, add a **Logo**.
4. Select **Submit** to create your integration.
5. Open the **Capabilities** tab. Select these capabilities:
 - Read content
 - Update content
 - Insert content
 - User information without email addresses
6. Be sure to **Save changes**.
7. Select the **Secrets** tab.
8. Copy the **Internal Integration Token** and add it as your n8n **Internal Integration Secret**.

Refer to the [Internal integration auth flow setup documentation](#) for more information about authenticating to the service.

Share Notion page(s) with the integration

For your integration to interact with Notion, you must [give your integration page permission](#) to interact with page(s) in your Notion workspace:

1. Visit the page in your Notion workspace.
2. Select the triple dot menu at the top right of a page.
3. In **Connections**, select **Connect to**.
4. Use the search bar to find and select your integration from the dropdown list.

Once you share at least one page with the integration, you can start making API requests. If the page isn't shared, any API requests made will respond with an error.

Refer to [Integration permissions](#) for more information.

Using OAuth2

To configure this credential, you'll need:

- A **Client ID**: Generated once you configure a public integration.
- A **Client Secret**: Generated once you configure a public integration.

You must [create a Notion integration](#) and set it to public distribution:

1. Go to your Notion [integration dashboard](#).
2. Select the **+ New integration** button.
3. Enter a **Name** for your integration, for example n8n integration. If desired, add a **Logo**.
4. Select **Submit** to create your integration.
5. Open the **Capabilities** tab. Select these capabilities:
 - Read content
 - Update content
 - Insert content
 - User information without email addresses

6. Select **Save changes**.
7. Go to the **Distribution** tab.
8. Turn on the **Do you want to make this integration public?** control.
9. Enter your company name and website in the **Organization Information** section.
10. Copy the n8n **OAuth Redirect URL** and add it to as a **Redirect URI** in the Notion integration's **OAuth Domain & URLs** section.
11. Go to the **Secrets** tab.
12. Copy the **Client ID** and **Client Secret** and add them to your n8n credential.

Refer to Notion's [public integration auth flow setup](#) for more information about authenticating to the service.

Internal vs. public integrations

Internal integrations are:

- Specific to a single workspace.
- Accessible only to members of that workspace.
- Ideal for custom workspace enhancements.

Internal integrations use a simpler authentication process (the integration secret) and don't require any security review before publishing.

Public integrations are:

- Usable across multiple, unrelated Notion workspaces.
- Accessible by any Notion user, regardless of their workspace.
- Ideal for catering to broad use cases.

Public integrations use the OAuth 2.0 protocol for authentication. They require a Notion security review before publishing.

For a more detailed breakdown of the two integration types, refer to Notion's [Internal vs. Public Integrations documentation](#).

npm credentials

You can use these credentials to authenticate the following nodes:

- [npm](#)

Prerequisites

Create an [npm](#) account.

Supported authentication methods

- API access token

Related resources

Refer to [npm's external integrations documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need:

- An **Access Token**: Create an access token by selecting **Access Tokens** from your profile menu. Refer to [npm's Creating and viewing access tokens documentation](#) for more detailed instructions.
 - A **Registry URL**: If you're using a custom npm registry, update the **Registry URL** to that custom registry. Otherwise, keep the public registry value.
-

Odoo credentials

You can use these credentials to authenticate the following nodes:

- [Odoo](#)

Supported authentication methods

- API key (Recommended)
- Password

Related resources

Refer to [Odoo's External API documentation](#) for more information about the service.

Refer to the Odoo [Getting Started tutorial](#) if you're new to Odoo.

Using API key

To configure this credential, you'll need a user account on an [Odoo](#) database and:

- Your **Site URL**
- Your **Username**
- An **API key**
- Your **Database name**

To set up the credential with an API key:

1. Enter your Odoo server or site URL as the **Site URL**.
2. Enter your **Username** as it's displayed on your **Change password** screen in Odoo.
3. To use an API key, go to **Your Profile > Preferences > Account Security > Developer API Keys**.

- If you don't have this option, you may need to upgrade your Odoo plan. Refer to [Required plan type](#) for more information.
4. Select **New API Key**.
 5. Enter a **Description** for the key, like n8n integration.
 6. Select **Generate Key**.
 7. Copy the key and enter it as the **Password or API key** in your n8n credential.
 8. Enter your Odoo **Database name**, also known as the instance name.

Refer to [Odoo API Keys](#) for more information.

Using password

To configure this credential, you'll need a user account on an [Odoo](#) database and:

- Your **Site URL**
- Your **Username**
- Your **Password**
- Your **Database name**

To set up the credential with a password:

1. Enter your Odoo server or site URL as the **Site URL**.
2. Enter your **Username** as it's displayed on your **Change password** screen in Odoo.
3. To use a password, enter your user password in the **Password or API key** field.
4. Enter your Odoo **Database name**, also known as the instance name.

Required plan type

Okta credentials

You can use these credentials to authenticate the following nodes:

- [Okta](#)

Prerequisites

Create an [Okta free trial](#) or create an admin account on an existing Okta org.

Supported authentication methods

- SSWS API Access token

Related resources

Refer to [Okta's documentation](#) for more information about the service.

Using SSWs API access token

To configure this credential, you'll need:

- The **URL**: The base URL of your Okta org, also referred to as your unique subdomain. There are two quick ways to access it:
 1. In the Admin Console, select your **Profile**, hover over the domain listed below your username, and select the **Copy** icon. Paste this into n8n, but be sure to add `https://` before it.
 2. Copy the base URL of your Admin Console URL, for example `https://dev-123456-admin.okta.com`. Paste it into n8n and remove `-admin`, for example: `https://dev-123456.okta.com`.
 - An **SSWS Access Token**: Create a token by going to **Security > API > Tokens > Create token**. Refer to [Create Okta API tokens](#) for more information.
-

Ollama credentials

You can use these credentials to authenticate the following nodes:

- [Ollama](#)
- [Chat Ollama](#)
- [Embeddings Ollama](#)

Prerequisites

Create and run an [Ollama](#) instance with one user. Refer to the [Ollama Quick Start](#) for more information.

Supported authentication methods

- Instance URL

Related resources

Refer to [Ollama's API documentation](#) for more information about the service.

-8<- “ snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Using instance URL

To configure this credential, you'll need:

- The **Base URL** of your Ollama instance or remote authenticated Ollama instances.
- (Optional) The **API Key** for Bearer token authentication if connecting to a remote, authenticated proxy.

The default **Base URL** is `http://localhost:11434`, but if you've set the `OLLAMA_HOST` environment variable, enter that value. If you have issues connecting to a local n8n server, try `127.0.0.1` instead of `localhost`.

If you're connecting to Ollama through authenticated proxy services (such as [Open WebUI](#)) you must include an API key. If you don't need authentication, leave this field empty. When provided, the API key is sent as a Bearer token in the `Authorization` header of the request to the Ollama API.

Refer to [How do I configure Ollama server?](#) for more information.

Ollama and self-hosted n8n

If you're self-hosting n8n on the same machine as Ollama, you may run into issues if they're running in different containers.

For this setup, open a specific port for n8n to communicate with Ollama by setting the `OLLAMA_ORIGINS` variable or adjusting `OLLAMA_HOST` to an address the other container can access.

Refer to Ollama's [How can I allow additional web origins to access Ollama?](#) for more information.

One Simple API credentials

You can use these credentials to authenticate the following nodes:

- [One Simple API](#)

Prerequisites

Create a [One Simple API](#) account.

Supported authentication methods

- API token

Related resources

Refer to [One Simple API's documentation](#) for more information about the service.

Using API token

To configure this credential, you'll need:

- An **API token**: Create a new API token on the [API Tokens](#) page. Be sure you select appropriate permissions for the token.

You can also access the API Tokens page by selecting your **Profile > API Tokens**.

Onfleet credentials

You can use these credentials to authenticate the following nodes:

- [Onfleet](#)
- [Onfleet Trigger](#)

Prerequisites

Create an [Onfleet](#) administrator account.

Supported authentication methods

- API key

Related resources

Refer to [Onfleet's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API key**: To create an API key, log into your organization's administrator account. Select **Settings > API & Webhooks**, then select **+** to create a new key. Refer to Onfleet's [Creating an API key documentation](#) for more information.
-

OpenAI credentials

You can use these credentials to authenticate the following nodes:

- [OpenAI](#)
- [Chat OpenAI](#)
- [Embeddings OpenAI](#)
- [LM OpenAI](#)

Prerequisites

Create an [OpenAI](#) account.

Supported authentication methods

- API key

Related resources

Refer to [OpenAI's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**
- An **Organization ID**: Required if you belong to multiple organizations; otherwise, leave this blank.

To generate your API Key:

1. Login to your OpenAI account or [create](#) an account.
2. Open your [API keys](#) page.
3. Select **Create new secret key** to create an API key, optionally naming the key.
4. Copy your key and add it as the **API Key** in n8n.

Refer to the [API Quickstart Account Setup documentation](#) for more information.

To find your Organization ID:

1. Go to your [Organization Settings](#) page.
2. Copy your Organization ID and add it as the **Organization ID** in n8n.

Refer to [Setting up your organization](#) for more information. Note that API requests made using an Organization ID will count toward the organization's subscription quota.

OpenCTI credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create an [OpenCTI](#) developer account.

Authentication methods

- API key

Related resources

Refer to [OpenCTI's documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows](#) and related content on n8n's website.

Using API key

To configure this credential, you'll need:

- An **API Key**: To get your API key, go to your **Profile > API access**. Refer to the OpenCTI [Integrations Authentication documentation](#) for more information.
-

OpenRouter credentials

You can use these credentials to authenticate the following nodes:

- [Chat OpenRouter](#)

Prerequisites

Create a [OpenRouter](#) account.

Supported authentication methods

- API key

Related resources

Refer to [OpenRouter's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**

To generate your API Key:

1. Login to your OpenRouter account or [create](#) an account.
2. Open your [API keys](#) page.
3. Select **Create new secret key** to create an API key, optionally naming the key.
4. Copy your key and add it as the **API Key** in n8n.

Refer to the [OpenRouter Quick Start](#) page for more information.

OpenWeatherMap credentials

You can use these credentials to authenticate the following nodes:

- [OpenWeatherMap](#)

Supported authentication methods

- API access token

Related resources

Refer to [OpenWeatherMap's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need an [OpenWeatherMap](#) account and:

- An **Access Token**

To get your **Access Token**:

1. After you verify your email address, OpenWeatherMap includes an **API Key** in your welcome email.
2. Copy that key and enter it in your n8n credential.

If you'd prefer to create a new key:

1. To create a new key, go to **Account > API Keys**.
 2. In the **Create Key** section, enter an **API Key Name**, like `n8n integration`.
 3. Select **Generate** to generate your key.
 4. Copy the generated key and enter it in your n8n credential.
-

Oracle Database credentials

You can use these credentials to authenticate the following nodes:

- [OracleDB](#)
- [Agent](#)

Prerequisites

Create a user account on an [Oracle Database](#) server.

Supported authentication methods

- Database connection

Related resources

Refer to [Oracle Database documentation](#) for more information about the service.

Using database connection

To configure this credential, you'll need:

- A **User** name.
- A **Password** for that user.
- **Connection String**: The Oracle Database instance to connect to. The string can be an Easy Connect string, or a TNS Alias from a tnsnames.ora file, or the Oracle Database instance.
- **Use Optional Oracle Client Libraries**: If you want to work with Oracle Database advanced features, turn this on. This option internally uses node-oracledb Thick mode. Additional settings to enable node-oracledb Thick mode are required. Refer to [Enabling Thick mode documentation](#) for more information. This option is not available in the official n8n docker images.
- **Use SSL**: If your Connection String is using SSL, turn this on and configure additional details for the SSL Authentication.
- **Wallet Password**: The password to decrypt the Privacy Enhanced Mail (PEM)-encoded private certificate, if it is encrypted.
- **Wallet Content**: The security credentials required to establish a mutual TLS (mTLS) connection to Oracle Database.
- **Distinguished Name**: The distinguished name (DN) that should be matched with the certificate DN.
- **Match Distinguished Name**: Whether the server certificate DN should be matched in addition to the regular certificate verification that is performed.
- **Allow Weak Distinguished Name Match**: Whether the secure DN matching behavior which checks both the listener and server certificates has to be performed.
- **Pool Min**: The number of connections established to the database when a pool is created.
- **Pool Max**: The maximum number of connections to which a connection pool can grow.
- **Pool Increment**: The number of connections that are opened whenever a connection request exceeds the number of currently open connections.
- **Pool Maximum Session Life Time**: The number of connections that are opened whenever a connection request exceeds the number of currently open connections.
- **Pool Connection Idle Timeout**: The number of connections that are opened whenever a connection request exceeds the number of currently open connections.
- **Connection Class Name**: DRCP/PRCP Connection Class. Refer to [Enabling DRCP](#) for more information.
- **Connection Timeout**: The timeout duration in seconds for an application to establish an Oracle Net connection.
- **Transport Connection Timeout**: The maximum number of seconds to wait to establish a connection to the database host.
- **Keepalive Probe Interval**: The number of minutes between the sending of keepalive probes.

To set up your database connection credential:

1. Enter your database's username as the **User** in your n8n credential.
2. Enter the user's **Password**.

3. Enter your database's connection string as the **Connection String** in your n8n credential.
4. If your database uses SSL and you'd like to configure **SSL** for the connection, turn this option on in the credential. If you turn it on, enter the information of your Oracle Database SSL certificate in these fields:
 1. Enter the wallet password, if any, in the **Wallet Password** field.
 2. Enter PEM-encoded wallet file, **ewallet.pem** contents in the 'Expanded' layout of the **Wallet Content** field. This will ensure that all the whitespaces from the PEM-encoded wallet file are retained. Direct copy-paste into the **Wallet Content** field will strip out the whitespaces and lead to connection errors.

Refer to [node-oracledb](#) for more information on working with TLS connections.

Oura credentials

You can use these credentials to authenticate the following nodes:

- [Oura](#)

Prerequisites

Create an [Oura](#) account.

Supported authentication methods

- API access token

Related resources

Refer to [Oura's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need:

- A **Personal Access Token**: To generate a personal access token, go to the [Personal Access Tokens](#) page and select **Create A New Personal Access Token**.

Refer to [How to Generate Personal Access Tokens](#) for more information.

Paddle credentials

You can use these credentials to authenticate the following nodes:

- [Paddle](#)

Prerequisites

Create a [Paddle](#) account.

Supported authentication methods

- API access token (Classic)

Related resources

Refer to [Paddle Classic's API documentation](#) for more information about the service.

Using API access token (Classic)

To configure this credential, you'll need:

- A **Vendor Auth Code**: Created when you generate an API key.
- A **Vendor ID**: Displayed when you generate an API key.
- **Use Sandbox Environment API**: When turned on, nodes using this credential will hit the Sandbox API endpoint instead of the live API endpoint.

To generate an auth code and view your Vendor ID, go to **Paddle > Developer Tools > Authentication > Generate Auth Code**. Select **Reveal Auth Code** to display the Auth Code. Refer to [API Authentication](#) for more information.

PagerDuty credentials

You can use these credentials to authenticate the following nodes:

- [PagerDuty](#)

Prerequisites

Create a [PagerDuty](#) account.

Supported authentication methods

- API token
- OAuth2

Related resources

Refer to [PagerDuty's API documentation](#) for more information about the service.

Using API token

To configure this credential, you'll need:

- A general access **API Token**: To generate an API token, go to **Integrations > Developer Tools > API Access Keys > Create New API Key**. Refer to [Generate a General Access REST API key](#) for more information.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you need to configure OAuth2 from scratch, [register a new Pagerduty app](#).

Use these settings for registering your app:

- In the **Category** dropdown list, select **Infrastructure Automation**.
- In the **Functionality** section, select **OAuth 2.0**.

Once you **Save** your app, open the app details and [edit your app configuration](#) to use these settings:

- Within the **OAuth 2.0** section, select **Add**.
- Copy the **OAuth Callback URL** from n8n and paste it into the **Redirect URL** field.
- Copy the **Client ID** and **Client Secret** from PagerDuty and add these to your n8n credentials.
- Select **Read/Write** from the **Set Permission Scopes** dropdown list.

Refer to the instructions in [App functionality](#) for more information on available functionality. Refer to the PagerDuty [OAuth Functionality documentation](#) for more information on the OAuth flow.

PayPal credentials

You can use these credentials to authenticate the following nodes:

- [PayPal](#)
- [PayPal Trigger](#)

Prerequisites

Create a [PayPal developer](#) account.

Supported authentication methods

- API client and secret

Related resources

Refer to [Paypal's API documentation](#) for more information about the service.

Using API client and secret

To configure this credential, you'll need:

- A **Client ID**: Generated when you create an app.
- A **Secret**: Generated when you create an app.
- An **Environment**: Select **Live** or **Sandbox**.

To generate the **Client ID** and **Secret**, log in to your Paypal [developer dashboard](#). Select **Apps & Credentials > Rest API apps > Create app**. Refer to [Get client ID and client secret](#) for more information.

Peekalink credentials

You can use these credentials to authenticate the following nodes:

- [Peekalink](#)

Prerequisites

Create a [Peekalink](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Peekalink's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: To get your API key, access your Peekalink [dashboard](#) and copy the key in the **Your API Key** section. Refer to [Get your API key](#) for more information.
-

Perplexity credentials

You can use these credentials to authenticate the following nodes:

- [Perplexity](#)

Supported authentication methods

- API key

Related resources

Refer to [Perplexity's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need a [Perplexity account](#) and:

- **a Perplexity API key:** You can find out how to create a Perplexity API key in the [Perplexity API getting started guide](#).

Refer to [Perplexity's API documentation](#) for more information about authenticating to the service.

PhantomBuster credentials

You can use these credentials to authenticate the following nodes:

- [PhantomBuster](#)

Prerequisites

Create a [PhantomBuster account](#).

Supported authentication methods

- API key

Related resources

Refer to [PhantomBuster's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- **An API Key:** To get an API key, go to [Workspace settings > Third party API keys](#) and select **+ Add API Key**. Refer to [How to find my API key](#) for more information.

Philips Hue credentials

You can use these credentials to authenticate the following nodes:

- [Philips Hue](#)

Prerequisites

Create a [Philips Hue](#) account.

Supported authentication methods

- OAuth2

Related resources

Refer to [Philips Hue's CLIP API documentation](#) for more information about the service.

Using OAuth2

-8<- “snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you're using the built-in OAuth connection, you don't need to enter an **APP ID**.

If you need to configure OAuth2 from scratch, you'll need a [Philips Hue developer](#) account

Create a new remote app on the [Add new Hue Remote API app](#) page.

Use these settings for your app:

- Copy the **OAuth Callback URL** from n8n and add it as a **Callback URL**.
 - Copy the **AppId**, **ClientId**, and **ClientSecret** and enter these in the corresponding fields in n8n.
-

Pinecone credentials

You can use these credentials to authenticate the following nodes:

- [Pinecone Vector Store](#)

Supported authentication methods

- API key

Related resources

Refer to [Pinecone's documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Using API key

To configure this credential, you'll need a [Pinecone](#) account and:

- An **API Key**

To get an API key:

1. Open your [Pinecone console](#).
2. Select the project you want to create an API key for. If you don't have any existing projects, create one. Refer to Pinecone's [Quickstart](#) for more information.
3. Go to [API Keys](#).
4. Copy the API Key displayed there and enter it in your n8n credential.

Refer to Pinecone's API [Authentication documentation](#) for more information.

Pipedrive credentials

You can use these credentials to authenticate the following nodes:

- [Pipedrive](#)
- [Pipedrive Trigger](#)

Supported authentication methods

- API token
- OAuth2

Related resources

Refer to [Pipedrive's developer documentation](#) for more information about the service.

Using API token

To configure this credential, you'll need a [Pipedrive](#) account and:

- An **API Token**

To get your API token:

1. Open your [API Personal Preferences](#).

2. Copy **Your personal API token** and enter it in your n8n credential.

If you have multiple companies, you'll need to select the correct company first:

1. Select your account name and be sure you're viewing the correct company.
2. Then select **Company Settings**.
3. Select **Personal Preferences**.
4. Select the **API** tab.
5. Copy **Your personal API token** and enter it in your n8n credential.

Refer to [How to find the API token](#) for more information.

Using OAuth2

To configure this credential, you'll need a [Pipedrive developer sandbox account](#) and:

- A **Client ID**
- A **Client Secret**

To get both, you'll need to register a new app:

1. Select your profile name in the upper right corner.
2. Find the company name of your sandbox account and select **Developer Hub**.
3. Select **Create an app**.
4. Select **Create public app**. The app's **Basic info** tab opens.
5. Enter an **App name** for your app, like n8n integration.
6. Copy the **OAuth Redirect URL** from n8n and add it as the app's **Callback URL**.
7. Select **Save**. The app's **OAuth & access scopes** tab opens.
8. Turn on appropriate **Scopes** for your app. Refer to [Pipedrive node scopes](#) and [Pipedrive Trigger node scopes](#) below for more guidance.
9. Copy the **Client ID** and enter it in your n8n credential.
10. Copy the **Client Secret** and enter it in your n8n credential.

Refer to [Registering a public app](#) for more information.

Pipedrive node scopes

The scopes you add to your app depend on which node(s) you want to use it for in n8n and what actions you want to complete with those.

Scopes you may need for the [Pipedrive](#) node:

| Object | Node action | UI scope | Actual scope |
|--------|-------------|----------|--------------|
|--------|-------------|----------|--------------|

| | | | |
|---------------|--|---|--|
| Activity | Get data of an activity
Get data of all activities | Activities:
Read only or
Activities:
Full Access | activities:read
or
activities:full |
| Activity | Create
Delete
Update | Activities:
Full Access | activities:full |
| Deal | Get data of a deal
Get data of all deals
Search a deal | Deals: Read only or
Deals: Full Access | deals:read or
deals:full |
| Deal | Create
Delete
Duplicate
Update | Deals: Full Access | deals:full |
| Deal Activity | Get all activities of a deal | Activities:
Read only or
Activities:
Full Access | activities:read
or
activities:full |
| Deal Product | Get all products in a deal | Products:
Read Only or
Products: Full Access | products:read
or
products:full |
| File | Download
Get data of a file | Refer to note below | Refer to note below |
| File | Create
Delete | Refer to note below | Refer to note below |
| Lead | Get data of a lead
Get data of all leads | Leads: Read only or
Leads: Full access | leads:read or
leads:full |
| Lead | Create
Delete
Update | Leads: Full access | leads:full |
| Note | Get data of a note
Get data of all notes | Refer to note below | Refer to note below |
| Note | Create
Delete
Update | Refer to note below | Refer to note below |
| Organization | Get data of an organization
Get data of all organizations
Search | Contacts:
Read Only or
Contacts: Full Access | contacts:read
or
contacts:full |
| Organization | Create
Delete
Update | Contacts: Full Access | contacts:full |
| Person | Get data of a person
Get data of all persons | Contacts:
Read Only or
Contacts: Full | contacts:read
or
contacts:full |

| | Search | Access |
|---------|----------------------------|--|
| Person | Create
Delete
Update | Contacts: Full Access contacts:full |
| Product | Get data of all products | Products: Read Only products:read |

The Pipedrive node also supports Custom API calls. Add relevant scopes for whatever custom API calls you intend to make.

Refer to [Scopes and permissions explanations](#) for more information.

Pipedrive Trigger node scopes

The [Pipedrive Trigger](#) node requires the **Webhooks: Full access** (webhooks:full) scope.

Plivo credentials

You can use these credentials to authenticate the following nodes:

- [Plivo](#)

Prerequisites

Create a [Plivo](#) account.

Supported authentication methods

- Basic auth

Related resources

Refer to [Plivo's API documentation](#) for more information about the service.

Using basic auth

To configure this credential, you'll need:

- An **Auth ID**: Acts like your username. Copy yours from the [Overview](#) page of the Plivo [console](#).
- An **Auth Token**: Acts like a password. Copy yours from the [Overview](#) page of the Plivo [console](#).

Refer to [How can I change my Auth ID or Auth Token?](#) for more detailed instructions.

Postgres credentials

You can use these credentials to authenticate the following nodes:

- [Postgres](#)
- [Agent](#)
- [Postgres Chat Memory](#)
- [PGVector Vector Store](#)

Prerequisites

[Create a user account](#) on a Postgres server.

Supported authentication methods

- Database connection

Related resources

Refer to [Postgres's documentation](#) for more information about the service.

Using database connection

To configure this credential, you'll need:

- The **Host** or domain name for the server.
- The **Database** name.
- A **User** name.
- A user **Password**.
- **Ignore SSL Issues**: Set whether the credential connects if SSL validation fails.
- **SSL**: Choose whether to use SSL in your connection.
- The **Port** number to use for the connection.
- **SSH Tunnel**: Choose if you want to use SSH to encrypt the network connection with the Postgres server.

To set up the database connection:

1. Enter the **Host** or domain name for the Postgres server. You can either run the `/conninfo` command to confirm the host name or run this query:

```
SELECT inet_server_addr();
```
2. Enter the **Database** name. Run the `/conninfo` command to confirm the database name.
3. Enter the **User** name of the user you wish to connect as.
4. Enter the user's **Password**.
5. **Ignore SSL Issues**: If you turn this on, the credential will connect even if SSL validation fails.

6. **SSL**: Choose whether to use SSL in your connection. Refer to Postgres [SSL Support](#) for more information. Options include:
 - **Allow**: Sets the `ssl-mode` parameter to `allow`. First try a non-SSL connection; if that fails, try an SSL connection.
 - **Disable**: Sets the `ssl-mode` parameter to `disable`. Only try a non-SSL connection.
 - **Require**: Sets the `ssl-mode` parameter to `require`. Only try an SSL connection. If a root CA file is present, verify that a trusted certificate authority (CA) issued the server certificate.

7. Enter the **Port** number to use for the connection. You can either run the `/conninfo` command to confirm the host name or run this query:

```
SELECT inet_server_port();
```

8. **SSH Tunnel**: Turn this setting on to connect to the database over SSH. Refer to [SSH tunnel limitations](#) for some guidance around using SSH. Once turned on, you'll need:

1. Select **SSH Authenticate with** to set the SSH Tunnel type to build:
 - Select **Password** if you want to connect to SSH using a password.
 - Select **Private Key** if you want to connect to SSH using an identity file (private key) and a passphrase.
2. Enter the remote bind address you're connecting to as the **SSH Host**.
3. **SSH Port**: Enter the local port number for the SSH tunnel.
4. **SSH Postgres Port**: Enter the remote end of the tunnel, the port number the database server is using.
5. **SSH User**: Enter the username to log in as.
6. If you selected **Password** for SSH Authenticate with, add the user's **SSH Password**.
7. If you selected **Private Key** for **SSH Authenticate with**:
 1. Add the contents of the **Private Key** or identity file used for SSH.
 2. If the **Private Key** was created with a passphrase, enter that **Passphrase**. If the **Private Key** has no passphrase, leave this field blank.

Refer to [Secure TCP/IP Connections with SSH Tunnels](#) for more information.

SSH tunnel limitations

Only use the **SSH Tunnel** setting if:

- You're using the credential with the [Postgres](#) node (Agent node doesn't support SSH tunnels).
- You have an SSH server running on the same machine as the Postgres server.
- You have a user account that can log in using `ssh`.

PostHog credentials

You can use these credentials to authenticate the following nodes:

- [PostHog](#)

Prerequisites

Create a [PostHog](#) account or host PostHog on your server.

Supported authentication methods

- API key

Related resources

Refer to [PostHog's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- The **API URL**: Enter the correct domain for your API requests:
 - On US Cloud, use <https://us.i.posthog.com> for public POST-only endpoints or <https://us.posthog.com> for private endpoints.
 - On EU Cloud, use <https://eu.i.posthog.com> for public POST-only endpoints or <https://eu.posthog.com> for private endpoints.
 - For self-hosted instances, use your self-hosted domain.
 - Confirm yours by checking your PostHog instance URL.
- An **API Key**: The API key you use depends on whether you're accessing public or private endpoints:
 - For public POST-only endpoints, use a [Project API key](#) from your project's **General** Settings.
 - For private endpoints, use a [Personal API key](#) from your User account's **Personal API Keys** Settings. Refer to [How to obtain a personal API key](#) for more information.

Postmark credentials

You can use these credentials to authenticate the following nodes:

- [Postmark Trigger](#)

Prerequisites

Create a [Postmark](#) account on a Postmark server.

Supported authentication methods

- API token

Related resources

Refer to [Postmark's API documentation](#) for more information about the service.

Using API token

To configure this credential, you'll need:

- A **Server API Token**: The Server API token is accessible by Account Owners, Account Admins, and users who have Server Admin privileges on a server. Get yours from the [API Tokens](#) tab under your Postmark server. Refer to [API Authentication](#) for more information.
-

ProfitWell credentials

You can use these credentials to authenticate the following nodes:

- [ProfitWell](#)

Prerequisites

Create a [ProfitWell](#) account.

Supported authentication methods

- API token

Related resources

Refer to [Profitwell's API documentation](#) for more information about the service.

Using API token

To configure this credential, you'll need:

- An **API Token**: To get an API key or token, go to **Account Settings > Integrations** and select **ProfitWell API**.
-

Pushbullet credentials

You can use these credentials to authenticate the following nodes:

- [Pushbullet](#)

Prerequisites

Create a [Pushbullet account](#).

Supported authentication methods

- OAuth2

Related resources

Refer to [Pushbullet's API documentation](#) for more information about the service.

Using OAuth2

To configure this credential, you'll need:

- A **Client ID**: Generated when you create a Pushbullet app, also known as an OAuth client.
- A **Client Secret**: Generated when you create a Pushbullet app, also known as an OAuth client.

To generate the **Client ID** and **Client Secret**, go to the [create client](#) page. Copy the **OAuth Redirect URL** from n8n and add this as your **redirect_uri** for the app/client. Use the **client_id** and **client_secret** from the OAuth Client in your n8n credential.

Refer to Pushbullet's [OAuth2 Guide](#) for more information.

Pushcut credentials

You can use these credentials to authenticate the following nodes:

- [Pushcut](#)
- [Pushcut Trigger](#)

Prerequisites

Download the [Pushcut app](#).

Supported authentication methods

- API key

Related resources

Refer to [Pushcut's Guides documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: To generate an API key, go to **Account > Integrations > Add API Key**. Refer to [Create an API key](#) for more information.
-

Pushover credentials

You can use these credentials to authenticate the following nodes:

- [Pushover](#)

Prerequisites

Create a [Pushover](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Pushover's API documentation](#) for more information about authenticating with the service.

Using API Key

To configure this credential, you'll need:

- An **API Key**: Generated when you [register an application](#). Refer to [Application Registration](#) for more information.
-

QRadar credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create a [QRadar](#) account.

Supported authentication methods

- API key

Related resources

Refer to [QRadar's documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n's website.

Using API key

To configure this credential, you'll need:

- An **API Key**: Also known as an authorized service token. Use the **Manage Authorized Services** window on the **Admin** tab to create an authentication token. Refer to [Creating an authentication token](#) for more information.
-

Qdrant credentials

You can use these credentials to authenticate the following nodes:

- [Qdrant Vector Store](#)

Supported authentication methods

- API key

Related resources

Refer to [Qdrant's documentation](#) for more information.

-8<- “ snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Using API key

To configure this credential, you'll need a [Qdrant cluster](#) and:

- An **API Key**
- Your **Qdrant URL**

To set it up:

1. Go to the [Cloud Dashboard](#).
2. Select **Access Management** to display available API keys (or go to the **API Keys** section of the **Cluster detail** page).
3. Select **Create**.
4. Select the cluster you want the key to have access to in the dropdown.
5. Select **OK**.
6. Copy the API Key and enter it in your n8n credential.
7. Enter the URL for your Qdrant cluster in the **Qdrant URL**. Refer

to [Qdrant Web UI](#) for more information.

Refer to [Qdrant's authentication documentation](#) for more information on creating and using API keys.

Qualys credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create a [Qualys](#) user account with any user role except Contact.

Supported authentication methods

- Basic auth

Related resources

Refer to [Qualys's documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n's website.

Using basic auth

To configure this credential, you'll need:

- A **Username**
 - A **Password**
 - A **Requested With** string: Enter a user description, like a user agent, or keep the default n8n application. This sets the required X-Requested-With header.
-

QuestDB credentials

You can use these credentials to authenticate the following nodes:

- [QuestDB](#)

Prerequisites

Create a user account on an instance of [QuestDB](#).

Supported authentication methods

- Database connection

Related resources

Refer to [QuestDB's documentation](#) for more information about the service.

Using database connection

To configure this credential, you'll need:

- The **Host**: Enter the host name or IP address for the server.
- The **Database**: Enter the database name, for example qdb.
- A **User**: Enter the username for the user account as configured in pg.user or pg.readonly.user property in server.conf. Default value is admin.
- A **Password**: Enter the password for the user account as configured in pg.password or pg.readonly.password property in server.conf. Default value is quest.
- **SSL**: Select whether the connection should use SSL, which sets the sslmode parameter. Options include:
 - Allow
 - Disable
 - Require
- The **Port**: Enter the port number to use for the connection. Default is 8812.

Refer to [List of supported connection properties](#) for more information.

Quick Base credentials

You can use these credentials to authenticate the following nodes:

- [Quick Base](#)

Prerequisites

Create a [Quick Base](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Quick Base's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- A **Hostname**: The string of characters located between `https://` and `/db` in your Quick Base URL.
 - A **User Token**: To generate a token, select your **Profile > My preferences > My User Information > Manage my user tokens**. Refer to [Creating and using user tokens](#) for detailed instructions.
-

QuickBooks credentials

You can use these credentials to authenticate the following nodes:

- [QuickBooks](#)

Prerequisites

Create an [Intuit developer](#) account.

Supported authentication methods

- OAuth2

Related resources

Refer to [Intuit's API documentation](#) for more information about the service.

Using OAuth2

To configure this credential, you'll need:

- A **Client ID**: Generated when you create an app.
- A **Client Secret**: Generated when you create an app.
- An **Environment**: Select whether this credential should access your **Production** or **Sandbox** environment.

To generate your **Client ID** and **Client Secret**, [create an app](#).

Use these settings when creating your app:

- Select appropriate scopes for your app. Refer to [Learn about scopes](#) for more information.
- Enter the **OAuth Redirect URL** from n8n as a **Redirect URI** in the app's **Development > Keys & OAuth** section.
- Copy the **Client ID** and **Client Secret** from the app's **Development > Keys & OAuth** section to enter in n8n. Refer to [Get the Client ID and Client Secret for your app](#) for more information.

Refer to Intuit's [Set up OAuth 2.0 documentation](#) for more information on the entire process.

RabbitMQ credentials

You can use these credentials to authenticate the following nodes:

- [RabbitMQ](#)
- [RabbitMQ Trigger](#)

Supported authentication methods

- User connection

Related resources

Refer to [RabbitMQ's Connections documentation](#) for more information about the service.

Using user connection

To configure this credential, you'll need to have a [RabbitMQ broker](#) installed and:

1. Enter the **Hostname** for the RabbitMQ broker.
2. Enter the **Port** the connection should use.
3. Enter a **User** the connection should use to log in as.
 - The default is guest. RabbitMQ recommends using a different user in production environments. Refer to [Access Control | The Basics](#) for more information. If you're using the guest account with a non-localhost connection, refer to [guest user issues](#) below for troubleshooting tips.
4. Enter the user's **Password**.
 - The default password for the guest user is guest.
5. Enter the virtual host the connection should use as the **Vhost**. The default virtual host is /.
6. Select whether the connection should use **SSL**. If turned on, also set:
 - **Passwordless**: Select whether the SSL certificate connection users SASL mechanism EXTERNAL (turned off) or doesn't use a password (turned on). If turned on, you'll also need to enter:
 - The **Client Certificate**: Paste the text of the SSL client certificate to use.
 - The **Client Key**: Paste the SSL client key to use.
 - The **Passphrase**: Paste the SSL passphrase to use.
 - **CA Certificates**: Paste the text of the SSL CA certificates to use.

guest user issues

If you use the guest user for the credential and you try to access a remote host, you may see a connection error. The RabbitMQ logs show an error like this:

```
[error] <0.918.0> PLAIN login refused: user 'guest' can only connect via localhost
```

This happens because RabbitMQ prohibits the default guest user from connecting from remote hosts. It can only connect over the `localhost`.

To resolve this error, you can:

- Update the guest user to allow it remote host access.
- Create or use a different user to connect to the remote host. The guest user is the only user limited by default.

Refer to [“guest” user can only connect from localhost](#) for more information.

Raindrop credentials

You can use these credentials to authenticate the following nodes:

- [Raindrop](#)

Prerequisites

Create a [Raindrop](#) account.

Supported authentication methods

- OAuth2

Related resources

Refer to [Raindrop’s API documentation](#) for more information about the service.

Using OAuth

To configure this credential, you’ll need:

- A **Client ID**
- A **Client Secret**

Generate both by creating a Raindrop app.

To create an app, go to **Settings > Integrations** and select **+ Create new app** in the **For Developers** section.

Use these settings for your app:

- Copy the **OAuth Redirect URL** from n8n and add it as a **Redirect URI** in your app.
 - Copy the **Client ID** and **Client Secret** from the Raindrop app and enter them in your n8n credential.
-

Rapid7 InsightVM credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create a [Rapid7 InsightVM](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Rapid7 InsightVM’s API documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n’s website.

Using API key

To configure this credential, you’ll need a [Rapid7 InsightVM](#) account and:

- A **URL**: The API endpoint URL where the resource or data you are requesting lives. You can find more information about the expected format in the [endpoint section of the Rapid7’s API overview](#).
- An **API Key**: Refer to [Rapid7’s Managing Platform API Keys documentation](#) to create an API key.

Refer to [Rapid7 InsightVM’s API documentation](#) for more information about authenticating to the service.

Recorded Future credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create a [Recorded Future](#) account.

Supported authentication methods

- API access token

Related resources

Refer to [Recorded Future's documentation](#) for more information about the service. The rest of Recorded Future's help center requires a paid account.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n's website.

Using API access token

To configure this credential, you'll need:

- An **API Access Token**

Refer to the [Recorded Future APIs documentation](#) for more information on getting your API access token.

Reddit credentials

You can use these credentials to authenticate the following nodes:

- [Reddit](#)

Prerequisites

Create a [Reddit](#) account.

Supported authentication methods

- OAuth2

Related resources

Refer to [Reddit's developer documentation](#) for more information about the service.

Using OAuth2

To configure this credential, you'll need:

- A **Client ID**
- A **Client Secret**

Generate both by creating a [third-party app](#). Visit the previous link or go to your **profile > Settings > Safety & Privacy > Manage third-party app authorization > are you a developer? create an app**.

Use these settings for your app:

- Copy the **OAuth Callback URL** from n8n and use it as your app's

redirect uri.

- The app's client ID displays underneath your app name. Copy that and add it as your n8n **Client ID**.
 - Copy the app's **secret** and add it as your n8n **Client Secret**.
-

Redis credentials

You can use these credentials to authenticate the following nodes:

- [Redis](#)
- [Redis Chat Memory](#)
- [Redis Vector Store](#)

Supported authentication methods

- Database connection

Related resources

Refer to [Redis's developer documentation](#) for more information about the service.

Using database connection

You'll need a user account on a [Redis](#) server and:

- A **Password**
- The **Host** name
- The **Port** number
- A **Database Number**
- **SSL**

To configure this credential:

1. Enter your user account **Password**.
2. Enter the **Host** name of the Redis server. The default is `localhost`.
3. Enter the **Port** number the connection should use. The default is `6379`.
 - This number should match the `tcp_port` listed when you run the `INFO` command.
4. Enter the **Database Number**. The default is `0`.
5. If the connection should use SSL, turn on the **SSL** toggle. If this toggle is off, the connection uses TCP only.
6. If you enable **SSL**, you have the option to **disable TLS verification**. Toggle to use self-signed certificates. **WARNING:** This makes the connection less secure.

Refer to [Connecting to Redis | Generic client](#) for more information.

Rocket.Chat credentials

You can use these credentials to authenticate the following nodes:

- [Rocket.Chat](#)

Prerequisites

- Create a [Rocket.Chat](#) account.
- Your account must have the `create-personal-access-tokens` permission to generate personal access tokens.

Supported authentication methods

- API access token

Related resources

Refer to [Rocket.Chat's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need:

- Your **User ID**: Displayed when you generate an access token.
- An **Auth Key**: Your personal access token. To generate an access token, go to your **avatar > Account > Personal Access Tokens**. Copy the token and add it as the n8n **Auth Key**.
- Your Rocket.Chat **Domain**: Also known as your default URL or workspace URL.

Refer to [Personal Access Tokens](#) for more information.

Rundeck credentials

You can use these credentials to authenticate the following nodes:

- [Rundeck](#)

Prerequisites

Create a user account on a [Rundeck](#) server.

Supported authentication methods

- API token

Related resources

Refer to [Rundeck's API documentation](#) for more information about the service.

Using API token

To configure this credential, you'll need:

- Your **URL**: Enter the base URL of your Rundeck server, for example `http://myserver:4440`. Refer to [URLs](#) for more information.
 - A user **API Token**: To generate a user API token, go to your **Profile > User API Tokens**. Refer to [User API tokens](#) for more information.
-

S3 credentials

You can use these credentials to authenticate the following nodes:

- [S3](#)

Prerequisites

Create an account on an S3-compatible server. Use the S3 node for generic or non-AWS S3 like:

- [DigitalOcean Spaces](#)
- [MinIO](#)
- [Wasabi](#)

Supported authentication methods

- S3 endpoint

Related resources

Refer to your S3-compatible provider's documentation for more information on the services. For example, refer to [Wasabi's REST API documentation](#) or [DigitalOcean's Spaces API Reference Documentation](#).

Using S3 endpoint

To configure this credential, you'll need:

- An **S3 Endpoint**: Enter the URL endpoint for the S3 storage backend.
- A **Region**: Enter the region for your S3 storage. Some providers call this the "region slug."
- An **Access Key ID**: Enter the S3 access key your S3 provider uses to access the bucket or space. Some providers call this API keys.
- A **Secret Access Key**: Enter the secret access key for the **Access Key ID**.

- **Force Path Style:** When turned on, the connection uses path-style addressing for buckets.
- **Ignore SSL Issues:** When turned on, n8n will connect even if SSL certificate validation fails.

More detailed instructions for DigitalOcean Spaces and Wasabi follow. If you're using a different provider, refer to their documentation for more information.

Using DigitalOcean Spaces

To configure the credential for use with DigitalOcean spaces:

1. In DigitalOceans, go to the control panel and open **Settings**. Your endpoint should be listed there. Prepend `https://` to that endpoint and enter it as the **S3 Endpoint** in n8n.
 - Your DigitalOceans endpoint depends on the data center region your bucket's in.
2. For the **Region**, enter the region your bucket's located in, for example, `nyc3`.
 - If you plan to use this credential to create new Spaces, enter `us-east-1` instead.
3. From your DigitalOceans control panel, go to [API](#).
4. Open the **Spaces Keys** tab.
5. Select **Generate New Key**.
6. Enter a **Name** for your key, like `n8n integration` and select the checkmark.
7. Copy the **Key** displayed next to the name and enter this as the **Access Key ID** in n8n.
8. Copy the **Secret** value and enter this as the **Secret Access Key** in n8n.
 - Refer to [Sharing Access to Buckets with Access Keys](#) for more information on generating the key and secret.
9. Keep the **Force Path Style** toggle turned off unless you want to use subdomain/virtual calling format.
10. Decide how you want the n8n credential to handle SSL:
 - To respect SSL certificate validation, keep the default of **Ignore SSL Issues** turned off.
 - To connect even if SSL certificate validation fails, turn on **Ignore SSL Issues**.

Refer to DigitalOcean's [Spaces API Reference Documentation](#) for more information.

Using Wasabi

To configure the credential for use with Wasabi:

1. For the **S3 Endpoint**, enter the service URL for your bucket's region. Start it with `https://`.
 - Refer to [Service URLs for Wasabi's Storage Regions](#) to identify the correct URL.
2. For the **Region**, enter the region slug portion of the service URL. For example, if you entered `https://s3.us-east-2.wasabisys.com` as the **S3 Endpoint**, `us-east-2` is the region.
3. Log into your Wasabi Console as the root user.
4. Open the **Menu** and select **Access Keys**.
5. Select **CREATE NEW ACCESS KEY**.
6. Select whether the key is for the **Root User** or a **Sub-User** and

- select **CREATE**.
7. Copy the **Access Key** and enter it in n8n as the **Access Key ID**.
 8. Copy the **Secret Key** and enter it in n8n as the **Secret Access Key**.
 - Refer to [Creating a New Access Key](#) for more information on generating the key and secret.
 9. Wasabi recommends turning on the **Force Path Style** toggle “because the path-style offers the greatest flexibility in bucket names, avoiding domain name issues.” Refer to the Wasabi [REST API Introduction](#) for more information.
 10. Decide how you want the n8n credential to handle SSL:
 - To respect SSL certificate validation, keep the default of **Ignore SSL Issues** turned off.
 - To connect even if SSL certificate validation fails, turn on **Ignore SSL Issues**.
-

Salesforce credentials

You can use these credentials to authenticate the following nodes:

- [Salesforce](#)
- [Salesforce trigger](#)

Supported authentication methods

- JWT
- OAuth2

Related resources

Refer to [Salesforce’s developer documentation](#) for more information about the service.

Using JWT

To configure this credential, you’ll need a [Salesforce](#) account and:

- Your **Environment Type** (Production or Sandbox)
- A **Client ID**: Generated when you create a connected app.
- Your Salesforce **Username**
- A **Private Key** for a self-signed digital certificate

To set things up, first you’ll create a private key and certificate, then a connected app:

1. In n8n, select the **Environment Type** for your connection. Choose the option that best describes your environment from **Production** or **Sandbox**.
2. Enter your Salesforce **Username**.
3. Log in to your org in Salesforce.
4. You’ll need a private key and certificate issued by a certification authority. Use your own key/cert or use OpenSSL to create a key and a self-signed digital certificate. Refer to the Salesforce [Create](#)

[a Private Key and Self-Signed Digital Certificate documentation](#) for instructions on creating your own key and certificate.

5. From **Setup** in Salesforce, enter App Manager in the Quick Find box, then select **App Manager**.
6. On the App Manager page, select **New Connected App**.
7. Enter the required **Basic Info** for your connected app, including a **Name** and **Contact Email address**. Refer to Salesforce's [Configure Basic Connected App Settings](#) documentation for more information.
8. Check the box to **Enable OAuth Settings**.
9. For the **Callback URL**, enter `http://localhost:1717/oauthRedirect`.
10. Check the box to **Use digital signatures**.
11. Select **Choose File** and upload the file that contains your digital certificate, such as `server.crt`.
12. Add these **OAuth scopes**:
 - **Full access (full)**
 - **Perform requests at any time (refresh_token, offline_access)**
13. Select **Save**, then **Continue**. The **Manage Connected Apps** page should open to the app you just created.
14. In the **API (Enable OAuth Settings)** section, select **Manage Consumer Details**.
15. Copy the **Consumer Key** and add it to your n8n credential as the **Client ID**.
16. Enter the contents of the private key file in n8n as **Private Key**.
 - Use the multi-line editor in n8n.
 - Enter the private key in standard PEM key format:
-----BEGIN PRIVATE KEY-----
KEY DATA GOES HERE
-----END PRIVATE KEY-----

These steps are what's required on the n8n side. Salesforce recommends setting refresh token policies, session policies, and OAuth policies too:

14. In Salesforce, select **Back to Manage Connected Apps**.
15. Select **Manage**.
16. Select **Edit Policies**.
17. Review the **Refresh Token Policy** field. Salesforce recommends using expire refresh token after 90 days.
18. In the **Session Policies** section, Salesforce recommends setting **Timeout Value** to 15 minutes.
19. In the **OAuth Policies** section, select **Admin approved users are pre-authorized for permitted users** for **Permitted Users**, and select **OK**.
20. Select **Save**.
21. Select **Manage Profiles**, select the profiles that are pre-authorized to use this connected app, and select **Save**.
22. Select **Manage Permission Sets** to select the permission sets. Create permission sets if necessary.

Refer to Salesforce's [Create a Connected App in Your Org](#) documentation for more information.

Using OAuth2

To configure this credential, you'll need a [Salesforce account](#).

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

Cloud and hosted users will need to select your **Environment Type**. Choose between **Production** and **Sandbox**.

If you’re self-hosting n8n, you’ll need to configure OAuth2 from scratch by creating a connected app:

1. In n8n, select the **Environment Type** for your connection. Choose the option that best describes your environment from **Production** or **Sandbox**.
2. Enter your Salesforce **Username**.
3. Log in to your org in Salesforce.
4. From **Setup** in Salesforce, enter App Manager in the Quick Find box, then select **App Manager**.
5. On the App Manager page, select **New Connected App**.
6. Enter the required **Basic Info** for your connected app, including a **Name** and **Contact Email address**. Refer to Salesforce’s Configure Basic Connected App Settings documentation for more information.
7. Check the box to **Enable OAuth Settings**.
8. For the **Callback URL**, enter `http://localhost:1717/OauthRedirect`.
9. Add these **OAuth scopes**:
 - **Full access (full)**
 - **Perform requests at any time (refresh_token, offline_access)**
10. Make sure the following settings are unchecked:
 - **Require Proof Key for Code Exchange (PKCE) Extension for Supported Authorization Flows**
 - **Require Secret for Web Server Flow**
 - **Require Secret for Refresh Token Flow**
11. Select **Save**, then **Continue**. The **Manage Connected Apps** page should open to the app you just created.
12. In the **API (Enable OAuth Settings)** section, select **Manage Consumer Details**.
13. Copy the **Consumer Key** and add it to your n8n credential as the **Client ID**.
14. Copy the **Consumer Secret** and add it to your n8n credential as the **Client Secret**.

These steps are what’s required on the n8n side. Salesforce recommends setting refresh token policies and session policies, too:

14. In Salesforce, select **Back to Manage Connected Apps**.
15. Select **Manage**.
16. Select **Edit Policies**.
17. Review the **Refresh Token Policy** field. Salesforce recommends using expire refresh token after 90 days.
18. In the **Session Policies** section, Salesforce recommends setting **Timeout Value** to 15 minutes.

Refer to Salesforce’s Create a Connected App in Your Org documentation for more information.

Common issues

Connection issues when authenticating with Salesforce from n8n Cloud

If you encounter connection issues when authenticating with Salesforce from n8n Cloud, you might need to enable a specific system permission in your Salesforce user profiles:

1. In Salesforce, go to **Setup**.
2. In the **Quick Find** box, search for **Profiles**.
3. Select the profile used by the user connecting to n8n (for example, System Administrator or the relevant profile).
4. Click **Edit** or use the new **Profile** interface if it's available.
5. Locate the **Administrative Permissions** section.
6. Enable the checkbox for **Approve Connected Apps for Non-Admins**. This checkbox might also appear as **Approve apps connected not installed** depending on your Salesforce language or translation.
7. Click **Save**.

This permission is not enabled by default, even for administrator profiles, and must be manually activated. Without this permission, you might experience authentication failures when trying to connect n8n to Salesforce.

Salesmate credentials

You can use these credentials to authenticate the following nodes:

- Salesmate

Prerequisites

Create a Salesmate account.

Supported authentication methods

- API token

Related resources

Refer to Salesmate's API documentation for more information about the service.

Using API token

To configure this credential, you'll need:

- A **Session Token**: An **Access Key**. Generate an access key in **My Account > Access Key**. Refer to Access Rights and Keys for more information.
- A **URL**: Your Salesmate domain name/base URL, for example `n8n.salesmate.io`.

SearXNG credentials

You can use these credentials to authenticate the following nodes:

- [SearXNG Tool](#)

Supported authentication methods

- API URL

Related resources

Refer to [SearXNG's documentation](#) for more information about the service.

Using API URL

To configure this credential, you'll need an instance of SearXNG running at an URL that's accessible from n8n:

- **API URL:** The URL of the SearXNG instance you want to connect to.

Refer to [SearXNG's Administrator documentation](#) for more information about running the service.

SeaTable credentials

You can use these credentials to authenticate the following nodes:

- [SeaTable](#)
- [SeaTable Trigger](#)

Prerequisites

Create a [SeaTable](#) account on either a cloud or self-hosted SeaTable server.

Supported authentication methods

- API key

Related resources

Refer to [SeaTable's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **Environment**: Select the environment that matches your SeaTable instance:
 - **Cloud-Hosted**
 - **Self-Hosted**
 - An **API Token (of a Base)**: Generate a **Base-Token** in SeaTable from the base options > **Advanced > API Token**.
 - Use **Read-Write** permission for your token.
 - Refer to [Creating an API token](#) for more information.
 - A **Timezone**: Select the timezone of your SeaTable server.
-

SecurityScorecard credentials

You can use these credentials to authenticate the following nodes:

- [SecurityScorecard](#)

Prerequisites

Create a [SecurityScorecard](#) account.

Supported authentication methods

- API key

Related resources

Refer to [SecurityScorecard's Developer documentation](#) and [API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Generate an API key in one of two ways:
 - As a user in [My Settings > API](#). Refer to [Get an API key](#) for more information.
 - As a bot user: View the bot user and select **create token**. Refer to [Authenticate with a bot user](#) for more information.
-

Segment credentials

You can use these credentials to authenticate the following nodes:

- [Segment](#)

Prerequisites

Create a [Segment](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Segment's Sources documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- A **Write Key**: To get a Write Key, go to **Sources > Add Source**. Add a **Node.js** source and copy that write key to add to your n8n credential.

Refer to [Locate your Write Key](#) for more information.

Sekoia credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create a [Sekoia SOC platform](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Sekoia's documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n's website.

Using API key

To configure this credential, you'll need:

- An **API Key**: To generate an API key, select **+ API Key**. Refer to [Create an API key](#) for more information.
-

Send Email credentials

You can use these credentials to authenticate the following nodes:

- [Send Email](#)

Prerequisites

- Create an email account on a service that supports SMTP.
- Some email providers require that you enable or set up outgoing SMTP or generate an app password. Refer to your provider's documentation to see if there are other required steps.

Supported authentication methods

- SMTP account

Related resources

Simple Message Transfer Protocol (SMTP) is a standard protocol for sending and receiving email. Most email providers offer instructions on setting up their service with SMTP. Refer to your provider's SMTP instructions.

Using SMTP account

To configure this credential, you'll need:

- A **User** email address
- A **Password**: This may be the user's password or an app password. Refer to the documentation for your email provider.
- The **Host**: The SMTP host address for your email provider, often formatted as `smtp.<provider>.com`. Check with your provider.
- A **Port** number: The port depends on the encryption method:
 - Port 465 for SSL/TLS (implicit encryption)
 - Port 587 for STARTTLS (explicit encryption)
 - Port 25 for no encryption (not recommended) Check with your email provider for their specific requirements.
- **SSL/TLS**: This toggle controls the encryption method:
 - Turn **ON** for port 465 (uses implicit SSL/TLS encryption)
 - Turn **OFF** for port 587 (uses STARTTLS explicit encryption)
 - Turn **OFF** for port 25 (no encryption)
- **Disable STARTTLS**: When SSL/TLS is disabled, the SMTP server can still try to upgrade the TCP connection using STARTTLS. Turning this on prevents that behaviour.
- **Client Host Name**: If needed by your provider, add a client host name. This name identifies the client to the server.

Provider instructions

Refer to the quickstart guides for these common email providers.

Gmail

Refer to [Gmail](#).

Outlook.com

Refer to [Outlook.com](#).

Yahoo

Refer to [Yahoo](#).

My provider isn't listed

If your email provider isn't listed here, search for **SMTP settings** to find their instructions. (These instructions may also be included with **IMAP settings** or **POP settings**.)

Gmail Send Email credentials

Follow these steps to configure the Send Email credentials with a Gmail account.

Prerequisites

To follow these instructions, you must first:

1. [Enable 2-step Verification](#) on your Gmail account.
2. [Generate an app password](#).

Enable 2-step Verification

-8<- “_snippets/integrations/builtin/credentials/email/gmail-two-step-verification.md”

Generate an app password

-8<- “_snippets/integrations/builtin/credentials/email/gmail-app-password.md”

Set up the credential

To set up the Send Email credential to use Gmail:

1. Enter your Gmail email address as the **User**.
2. Enter the app password you generated above as the **Password**.
3. Enter `smtp.gmail.com` as the **Host**.
4. For the **Port**:

- Keep the default 465 for SSL or if you’re unsure what to use.
 - Enter 587 for TLS.
5. Turn on the **SSL/TLS** toggle.

Refer to the Outgoing Mail (SMTP) Server settings in [Read Gmail messages on other email clients using POP](#) for more information. If the settings above don’t work for you, check with your email administrator.

Outlook.com Send Email credentials

Follow these steps to configure the Send Email credentials with an Outlook.com account.

Set up the credential

To configure the Send Email credential to use an Outlook.com account:

1. Enter your Outlook.com email address as the **User**.
2. Enter your Outlook.com password as the **Password**.
3. Enter `smtp-mail.outlook.com` as the **Host**.
4. Enter 587 for the **Port**.
5. Turn on the **SSL/TLS** toggle.

Refer to Microsoft’s [POP, IMAP, and SMTP settings for Outlook.com](#) documentation for more information. If the settings above don’t work for you, check with your email administrator.

Use an app password

-8<- “`_snippets/integrations/builtin/credentials/email/outlook-app-password.md`”

Yahoo Send Email credentials

Follow these steps to configure the Send Email credentials with a Yahoo account.

Prerequisites

To follow these instructions, you must first generate an app password:

-8<- “`_snippets/integrations/builtin/credentials/email/yahoo-app-password.md`”

Set up the credential

To configure the Send Email credential to use Yahoo Mail:

1. Enter your Yahoo email address as the **User**.
2. Enter the app password you generated above as the **Password**.
3. Enter `smtp.mail.yahoo.com` as the **Host**.
4. For the **Port**:
 - Keep the default 465 for SSL or if you're unsure what to use.
 - Enter 587 for TLS.
5. Turn on the **SSL/TLS** toggle.

Refer to [IMAP server settings for Yahoo Mail](#) for more information. If the settings above don't work for you, check with your email administrator.

SendGrid credentials

You can use these credentials to authenticate the following nodes:

- [SendGrid](#)

Supported authentication methods

- API key

Related resources

Refer to [SendGrid's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need a [SendGrid](#) account and:

- An **API Key**

To create an API key:

1. In the Twilio SendGrid app, go to **Settings > API Keys**.
2. Select **Create API Key**.
3. Enter a **Name** for your API key, like `n8n integration`.
4. Select **Full Access**.
5. Select **Create & View**.
6. Copy the key and enter it in your n8n credential.

Refer to [Create API Keys](#) for more information.

Sendy credentials

You can use these credentials to authenticate the following nodes:

- [Sendy](#)

Prerequisites

Host a [Sendy](#) application.

Supported authentication methods

- API key

Related resources

Refer to [Sendy's API documentation](#) for more information about the service.

Using API Key

To configure this credential, you'll need:

- A **URL**: The URL of your Sendy application.
 - An **API Key**: Get your API key from your user profile > **Settings** > **Your API Key**.
-

Sentry.io credentials

You can use these credentials to authenticate the following nodes:

- [Sentry.io](#)

Prerequisites

Create a [Sentry.io](#) account.

Supported authentication methods

- API token
- OAuth2
- Server API token: Use for [self-hosted Sentry](#).

Related resources

Refer to [Sentry.io's API documentation](#) for more information about the service.

Using API token

To configure this credential, you'll need:

- An API **Token**: Generate a User Auth Token in **Account > Settings > User Auth Tokens**. Refer to User Auth Tokens for more information.

Using OAuth

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you need to configure OAuth2 from scratch, create an integration with these settings:

- Copy the n8n **OAuth Callback URL** and add it as an **Authorized Redirect URI**.
- Copy the **Client ID** and **Client Secret** and add them to your n8n credential.

Refer to Public integrations for more information on creating the integration.

Using Server API token

To configure this credential, you'll need:

- An API **Token**: Generate a User Auth Token in **Account > Settings > User Auth Tokens**. Refer to User Auth Tokens for more information.
 - The **URL** of your self-hosted Sentry instance.
-

Serp credentials

You can use these credentials to authenticate the following nodes:

- Serp

Prerequisites

Create a SerpApi account.

Supported authentication methods

- API key

Related resources

Refer to Serp's API documentation for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Using API key

To configure this credential, you'll need:

- An **API Key**

To get your API key:

1. Go to **Your Account > API Key**.
 2. Copy **Your Private API Key** and enter it as the **API Key** in your n8n credential.
-

ServiceNow credentials

You can use these credentials to authenticate the following nodes:

- ServiceNow

Prerequisites

Create a ServiceNow developer account.

Supported authentication methods

- Basic auth
- OAuth2

Related resources

Refer to ServiceNow's API documentation for more information about the service.

Using basic auth

To configure this credential, you'll need:

- A **User** name: Enter your ServiceNow username.
- A **Password**: Enter your ServiceNow password.
- A **Subdomain**: The subdomain for your servicenow instance is in your instance URL: `https://<subdomain>.service-now.com/`. For example, if the full URL is `https://dev99890.service-now.com`, then the subdomain is dev99890.

Using OAuth2

To configure this credential, you'll need:

- A **Client ID**: Generated once you register a new app.
- A **Client Secret**: Generated once you register a new app.
- A **Subdomain**: The subdomain for your servicenow instance is in your instance URL: `https://<subdomain>.service-now.com/`. For

example, if the full URL is <https://dev99890.service-now.com>, then the subdomain is dev99890.

To generate your **Client ID** and **Client Secret**, register a new app in **System OAuth > Application Registry > New > Create an OAuth API endpoint for external clients**. Use these settings for your app:

- Copy the **Client ID** and add it to your n8n credential.
- Enter a **Client Secret** or leave it blank to automatically generate a random secret. Add this secret to your n8n credential.
- Copy the n8n **OAuth Redirect URL** and add it as a **Redirect URL**.

Refer to [How to setup OAuth2 authentication for RESTMessageV2 integrations](#) for more information.

seven credentials

You can use these credentials to authenticate the following nodes:

- [seven](#)

Prerequisites

Create a [seven](#) developer account.

Supported authentication methods

- API key

Related resources

Refer to [seven's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API key**: Go to **Account > Developer > API Keys** to create an API key. Refer to [API First Steps](#) for more information.
-

Shopify credentials

You can use these credentials to authenticate the following nodes with Shopify.

- [Shopify](#)
- [Shopify Trigger](#)

Supported authentication methods

- Access token (recommended): For private apps/single store use. Can be created by regular admins.
- OAuth2: For public apps. Must be created by partner accounts.
- API key: Deprecated.

Related resources

Refer to [Shopify's authentication documentation](#) for more information about the service.

Using access token

To configure this credential, you'll need a [Shopify](#) admin account and:

- Your **Shop Subdomain**
- An **Access Token**: Generated when you create a custom app.
- An **APP Secret Key**: Generated when you create a custom app.

To set up the credential, you'll need to create and install a custom app:

1. Enter your **Shop Subdomain**.
 - Your subdomain is within the URL:
`https://<subdomain>.myshopify.com`. For example, if the full URL is `https://n8n.myshopify.com`, the Shop Subdomain is n8n.
2. In Shopify, go to **Admin > Settings > Apps and sales channels**.
3. Select **Develop apps**.
4. Select **Create a custom app**.
5. In the modal window, enter the **App name**.
6. Select an **App developer**. The app developer can be the store owner or any account with the **Develop apps** permission.
7. Select **Create app**.
8. Select **Select scopes**. In the **Admin API access scopes** section, select the API scopes you want for your app.
 - To use all functionality in the **Shopify** node, add the `read_orders`, `write_orders`, `read_products`, and `write_products` scopes.
 - Refer to [Shopify API Access Scopes](#) for more information on the available scopes.
9. Select **Save**.
10. Select **Install app**.
11. In the modal window, select **Install app**.
12. Open the app's **API Credentials** section.

13. Copy the **Admin API Access Token**. Enter this in your n8n credential as the **Access Token**.
14. Copy the **API Secret Key**. Enter this in your n8n credential as the **APP Secret Key**.

Refer to [Creating a custom app](#) and [Generate access tokens for custom apps in the Shopify admin](#) for more information on these steps.

Using OAuth2

To configure this credential, you'll need a [Shopify partner](#) account and:

- A **Client ID**: Generated when you create a custom app.
- A **Client Secret**: Generated when you create a custom app.
- Your **Shop Subdomain**

To set up the credential, you'll need to create and install a custom app:

1. Open your [Shopify Partner dashboard](#).
2. Select **Apps** from the left navigation.
3. Select **Create app**.
4. In the **Use Shopify Partners** section, enter an **App name**.
5. Select **Create app**.
6. When the app details open, copy the **Client ID**. Enter this in your n8n credential.
7. Copy the **Client Secret**. Enter this in your n8n credential.
8. In the left menu, select **Configuration**.
9. In n8n, copy the **OAuth Redirect URL** and paste it into the **Allowed redirection URL(s)** in the **URLs** section.
10. In the **URLs** section, enter an **App URL** for your app. The host entered here needs to match the host for the **Allowed redirection URL(s)**, like the base URL for your n8n instance.
11. Select **Save and release**.
12. Select **Overview** from the left menu. At this point, you can choose to **Test your app** by installing it to one of your stores, or **Choose distribution** to distribute it publicly.
13. In n8n, enter the **Shop Subdomain** of the store you installed the app to, either as a test or as a distribution.
 - Your subdomain is within the URL:
<https://<subdomain>.myshopify.com>. For example, if the full URL is <https://n8n.myshopify.com>, the Shop Subdomain is n8n.

Using API key

To configure this credential, you'll need:

- An **API Key**
- A **Password**
- Your **Shop Subdomain**: Your subdomain is within the URL:
<https://<subdomain>.myshopify.com>. For example, if the full URL is <https://n8n.myshopify.com>, the Shop Subdomain is n8n.
- *Optional:* A **Shared Secret**

Common issues

Here are some common issues setting up the Shopify credential and steps to resolve or troubleshoot them.

Enable custom app development

If you don't see the option to **Create a custom app**, no one's enabled custom app development for your store.

To enable custom app development, you must log in either as a store owner or as a user with the **Enable app development** permission:

1. In Shopify, go to **Admin > Settings > Apps and sales channels**.
2. Select **Develop apps**.
3. Select **Allow custom app development**.
4. Read the warning and information provided and select **Allow custom app development**.

Forbidden credentials error

If you get a **Couldn't connect with these settings / Forbidden - perhaps check your credentials** warning when you test the credentials, this may be due to your app's `access_scope` dependencies. For example, the `read_orders` scope also requires `read_products` scope. Review the scopes you have assigned and the action you're trying to complete.

Shuffler credentials

```
-8<- "_snippets/integrations/builtin/credentials/cred-only-statement.md"
```

Prerequisites

Create a [Shuffler](#) account on either a cloud or self-hosted instance.

Supported authentication methods

- API key

Related resources

Refer to [Shuffler's documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n's website.

Using API key

To configure this credential, you'll need:

- An **API Key**: Get your API key from the **Settings** page.
-

SIGNL4 credentials

You can use these credentials to authenticate the following nodes:

- [SIGNL4](#)

Prerequisites

Create a [SIGNL4](#) account.

Supported authentication methods

- Webhook secret

Related resources

Refer to [SIGNL4's Inbound Webhook](#) documentation for more information about the service.

Using webhook secret

To configure this credential, you'll need:

- A **Team Secret**: SIGNL4 includes this secret in the “ Sign up complete” email as the last part of the webhook URL. If your webhook URL is `https://connect.signl4.com/webhook/helloworld`, your team secret would be `helloworld`.
-

Slack credentials

You can use these credentials to authenticate the following nodes:

- [Slack](#)
- [Slack Trigger](#)

Supported authentication methods

- API access token:
 - Required for the [Slack Trigger](#) node.
 - Works with the [Slack](#) node, but not recommended.
- OAuth2:
 - Recommended method for the [Slack](#) node.
 - Doesn't work with the [Slack Trigger](#) node.

Related resources

Refer to [Slack's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need a [Slack](#) account and:

- An **Access Token**

To generate an access token, create a Slack app:

1. Open your [Slack API Apps](#) page.
2. Select **Create New App > From scratch**.
3. Enter an **App Name**.
4. Select the **Workspace** where you'll be developing your app.
5. Select **Create App**. The app details open.
6. In the left menu under **Features**, select **OAuth & Permissions**.
7. In the **Scopes** section, select appropriate scopes for your app. Refer to [Scopes](#) for a list of recommended scopes.
8. After you've added scopes, go up to the **OAuth Tokens** section and select **Install to Workspace**. You must be a Slack workspace admin to complete this action.
9. Select **Allow**.
10. Copy the **Bot User OAuth Token** and enter it as the **Access Token** in your n8n credential.
11. If you're using this credential for the [Slack Trigger](#), follow the steps in [Slack Trigger configuration](#) to finish setting up your app.

Refer to the Slack API [Quickstart](#) for more information.

Slack Trigger configuration

To use your Slack app with the [Slack Trigger](#) node:

1. Go to [Your Apps](#) in Slack and select the app you want to use.
2. Go to **Features > Event Subscriptions**.
3. Turn on the **Enable Events** control.
4. In n8n, copy the **Webhook URL** and enter it as the **Request URL** in your Slack app.
5. Once verified, select the bot events to subscribe to. Use the **Trigger on** field in n8n to filter these requests.
 - To use an event not in the list, add it as a bot event and select **Any Event** in the n8n node.

Refer to [Quickstart | Configuring the app for event listening](#) for more information.

n8n recommends enabling request signature verification for your Slack Trigger for additional security:

1. Go to [Your Apps](#) in Slack and select the app you want to use.
2. Go to **Settings > Basic Information**.

3. Copy the value of **Signing**.
4. In n8n, Paste this value into the **Signature Secret** field for the credential.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you’re self-hosting n8n and need to configure OAuth2 from scratch, you’ll need a Slack account and:

- A **Client ID**
- A **Client Secret**

To get both, create a Slack app:

1. Open your Slack API Apps page.
2. Select **Create New App > From scratch**.
3. Enter an **App Name**.
4. Select the **Workspace** where you’ll be developing your app.
5. Select **Create App**. The app details open.
6. In **Settings > Basic Information**, open the **App Credentials** section.
7. Copy the **Client ID** and **Client Secret**. Paste these into the corresponding fields in n8n.
8. In the left menu under **Features**, select **OAuth & Permissions**.
9. In the **Redirect URLs** section, select **Add New Redirect URL**.
10. Copy the **OAuth Callback URL** from n8n and enter it as the new Redirect URL in Slack.
11. Select **Add**.
12. Select **Save URLs**.
13. In the **Scopes** section, select appropriate scopes for your app. Refer to Scopes for a list of scopes.
14. After you’ve added scopes, go up to the **OAuth Tokens** section and select **Install to Workspace**. You must be a Slack workspace admin to complete this action.
15. Select **Allow**.
16. At this point, you should be able to select the OAuth button in your n8n credential to connect.

Refer to the Slack API Quickstart for more information. Refer to the Slack Installing with OAuth documentation for more details on the OAuth flow itself.

Scopes

Scopes determine what permissions an app has.

- If you want your app to act on behalf of users who authorize the app, add the required scopes under the **User Token Scopes** section.
- If you’re building a bot, add the required scopes under the **Bot Token Scopes** section.

Here’s the list of scopes the OAuth credential requires, which are a good starting point:

| Scope name | Notes |
|---------------------|------------------------------------|
| channels:read | |
| channels:write | Not available as a bot token scope |
| channels:history | |
| chat:write | |
| files:read | |
| files:write | |
| groups:read | |
| groups:history | |
| im:read | |
| im:history | |
| mpim:read | |
| mpim:history | |
| reactions:read | |
| reactions:write | |
| stars:read | Not available as a bot token scope |
| stars:write | Not available as a bot token scope |
| usergroups:read | |
| usergroups:write | |
| users.profile:read | |
| users.profile:write | Not available as a bot token scope |
| users:read | |
| search:read | |

Common issues

Token expired

-8<- “_snippets/integrations/builtin/credentials/slack/token-rotation.md”

Snowflake credentials

You can use these credentials to authenticate the following nodes:

- [Snowflake](#)

Prerequisites

Create a [Snowflake](#) account.

Supported authentication methods

- Database connection

Related resources

Refer to [Snowflake's API documentation](#) and [SQL Command Reference](#) for more information about the service.

Using database connection

To configure this credential, you'll need:

- An **Account**: Your account name is the string of characters located between `https://` and `snowflakecomputing.com` in your Snowflake URL. For example, if the URL of your Snowflake account is `https://abc.eu-central-1.snowflakecomputing.com` then the name of your account is `abc.eu-central-1`.
- A **Database**: Enter the name of the database the credential should connect to.
- A **Warehouse**: Enter the name of the default virtual warehouse to use for the session after connecting. n8n uses this warehouse for performing queries, loading data, and so on.
- A **Username**
- A **Password**
- A **Schema**: Enter the schema you want to use after connecting.
- A **Role**: Enter the security role you want to use after connecting.
- **Client Session Keep Alive**: By default, client connections typically time out three or four hours after the most recent query execution. Turning this setting on sets the `clientSessionKeepAlive` parameter to true: the server will keep the client's connection alive indefinitely, even if the connection doesn't execute any queries.

Refer to [Session Commands](#) for more information on these settings.

SolarWinds IPAM credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Supported authentication methods

- Username & Password

Related resources

Refer to [SolarWinds IPAM's API documentation](#) for more information about the service.

Using Username & Password

To configure this credential, you'll need a SolarWinds IPAM account and:

- **URL**: The base URL of your SolarWinds IPAM server

- **Username:** The username you use to access SolarWinds IPAM
- **Password:** The password you use to access SolarWinds IPAM

Refer to [SolarWinds IPAM's API documentation](#) for more information about authenticating to the service.

SolarWinds Observability SaaS credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Supported authentication methods

- API Token

Related resources

Refer to [SolarWinds Observability SaaS's API documentation](#) for more information about the service.

Using API Token

To configure this credential, you'll need a SolarWinds Observability SaaS account and:

- **URL:** The URL you use to access the SolarWinds Observability SaaS platform
- **API Token:** An API token found in the SolarWinds Observability SaaS platform under **Settings > Api Tokens**

Refer to [SolarWinds Observability SaaS's API documentation](#) for more information about authenticating to the service.

Splunk credentials

You can use these credentials to authenticate the following nodes:

- [Splunk](#)

Prerequisites

- [Download and install](#) Splunk Enterprise.
- [Enable token authentication](#) in **Settings > Tokens**.

Supported authentication methods

- API auth token

Related resources

Refer to [Splunk's Enterprise API documentation](#) for more information about the service.

Using API auth token

To configure this credential, you'll need:

- An **Auth Token**: Once you've enabled token authentication, create an auth token in **Settings > Tokens**. Refer to [Creating authentication tokens](#) for more information.
- A **Base URL**: For your Splunk instance. This should include the protocol, domain, and port, for example: <https://localhost:8089>.
- **Allow Self-Signed Certificates**: If turned on, n8n will connect even if SSL validation fails.

Required capabilities

Your Splunk platform account and role must have certain capabilities to create authentication tokens:

- `edit_tokens_own`: Required if you want to create tokens for yourself.
- `edit_tokens_all`: Required if you want to create tokens for any user on the instance.

Refer to [Define roles on the Splunk platform with capabilities](#) for more information.

Spotify credentials

You can use these credentials to authenticate the following nodes:

- [Spotify](#)

Supported authentication methods

- OAuth2

Related resources

Refer to [Spotify's Web API documentation](#) for more information about the service.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you're self-hosting n8n, you'll need a Spotify Developer account so you can create a Spotify app:

1. Open the Spotify developer dashboard.
2. Select **Create an app**.
3. Enter an **App name**, like n8n integration.
4. Enter an **App description**.
5. Copy the **OAuth Redirect URL** from n8n and enter it as the **Redirect URI** in your Spotify app.
6. Check the box to agree to the Spotify Terms of Service and Branding Guidelines.
7. Select **Create**. The **App overview** page opens.
8. Copy the **Client ID** and enter it in your n8n credential.
9. Copy the **Client Secret** and enter it in your n8n credential.
10. Select **Connect my account** and follow the on-screen prompts to finish authorizing the credential.

Refer to Spotify Apps for more information.

SSH credentials

You can use these credentials to authenticate the following nodes:

- SSH

Prerequisites

- Create a remote server with SSH enabled.
- Create a user account that can ssh into the server using one of the following:
 - Their own password
 - An SSH private key

Supported authentication methods

- Password: Use this method if you have a user account that can ssh into the server using their own password.
- Private key: Use this method if you have a user account that uses an SSH key for the server or service.

Related resources

Secure Shell (SSH) protocol is a method for securely sending commands over a network. Refer to Connecting to GitHub with SSH for an example of SSH setup.

Using password

Use this method if you have a user account that can ssh into the server using their own password.

To configure this credential, you'll need to:

1. Enter the IP address of the server you're connecting to as the **Host**.
2. Enter the **Port** to use for the connection. SSH uses port 22 by default.
3. Enter the **Username** for a user account with ssh access on the server.
4. Enter the **Password** for that user account.

Using private key

Use this method if you have a user account that uses an SSH key for the server or service.

To configure this credential, you'll need to:

1. Enter the IP address of the server you're connecting to as the **Host**.
2. Enter the **Port** to use for the connection. SSH uses port 22 by default.
3. Enter the **Username** of the account that generated the private key.
4. Enter the entire contents of your SSH **Private Key**.
5. If you created a **Passphrase** for the **Private Key**, enter the passphrase.
 - If you didn't create a passphrase for the key, leave blank.

Stackby credentials

You can use these credentials to authenticate the following nodes:

- [Stackby](#)

Prerequisites

Create a [Stackby](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Stackby's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Go to your [Account Settings > API](#) to create an API Key. Refer to [API Key](#) for more information.

Storyblok credentials

You can use these credentials to authenticate the following nodes:

- [Storyblok](#)

Prerequisites

Create a [Storyblok](#) account.

Supported authentication methods

- Content API key: For read-only access
- Management API key: For full CRUD operations

Related resources

Refer to Storyblok's [Content v1 API documentation](#) and [Management API documentation](#) for more information about the services.

Using Content API key

To configure this credential, you'll need:

- A **Content API Key**: Go to your Storyblok workspace's **Settings > Access Tokens** to get an API key. Choose an **Access Level** of either **Public** (version=published) or **Preview** (version-published and version=draft). Enter this access token as your **API Key**. Refer to [How to retrieve and generate access tokens](#) for more detailed instructions.

Refer to [Content v1 API Authentication](#) for more information about supported operations with each Access Level.

Using Management API key

To configure this credential, you'll need:

- A **Personal Access Token**: Go to **My Account > Personal access tokens** to generate a new access token. Enter this access token as your **Personal Access Token**.

Strapi credentials

You can use these credentials to authenticate the following nodes:

- [Strapi](#)

Prerequisites

Create a [Strapi](#) admin account with:

- Access to an existing Strapi project.
- At least one collection type within that project.
- Published data within that collection type.

Refer to the Strapi developer [Quick Start Guide](#) for more information.

Supported authentication methods

- API user account: Requires a user account with appropriate content permissions.
- API token: Requires an admin account.

Related resources

Refer to [Strapi's documentation](#) for more information about the service.

Using API user account

To configure this credential, you'll need:

- A user **Email**: Must be for a user account, not an admin account. Refer to the more detailed instructions below.
- A user **Password**: Must be for a user account, not an admin account. Refer to the more detailed instructions below.
- The **URL**: Use the public URL of your Strapi server, defined in `./config/server.js` as the `url` parameter. Strapi recommends using an absolute URL.
 - For Strapi Cloud projects, use the URL of your Cloud project, for example: `https://my-strapi-project-name.strapiapp.com`
- The **API Version**: Select the version of the API you want your calls to use. Options include:
 - **Version 3**
 - **Version 4**

In Strapi, the configuration involves two steps:

1. [Configure a role](#).
2. [Create a user account](#).

Refer to the more detailed instructions below for each step.

Configure a role

For API access, use the Users & Permissions Plugin in **Settings > Users & Permissions Plugin**.

Refer to [Configuring Users & Permissions Plugin](#) for more information on the plugin. Refer to [Configuring end-user roles](#) for more information on roles.

For the n8n credential, the user must have a role that grants them API permissions on the collection type. For the role, you can either:

- Update the default **Authenticated** role to include the permissions and assign the user to that role. Refer to [Configuring role's permissions](#) for more information.
- Create a new role to include the permissions and assign the user to that role. Refer to [Creating a new role](#) for more information.

For either option, once you open the role:

1. Go to the **Permissions** section.
2. Open the section for the relevant collection type.
3. Select the permissions for the collection type that the role should have. Options include:
 - `create` (POST)
 - `find` and `findone` (GET)
 - `update` (PUT)
 - `delete` (DELETE)
4. Repeat for all relevant collection types.
5. Save the role.

Refer to [Endpoints](#) for more information on the permission options.

Create a user account

Now that you have an appropriate role, create an end-user account and assign the role to it:

1. Go to **Content Manager > Collection Types > User**.
2. Select **Add new entry**.
3. Fill in the user details. The n8n credential requires these fields, though your Strapi project may have more custom required fields:
 - **Username**: Required for all Strapi users.
 - **Email**: Enter in Strapi and use as the **Email** in the n8n credential.
 - **Password**: Enter in Strapi and use as the **Password** in the n8n credential.
 - **Role**: Select the role you set up in the previous step.

Refer to [Managing end-user accounts](#) for more information.

Using API token

To configure this credential, you'll need:

- An **API Token**: Create an API token from **Settings > Global Settings > API Tokens**. Refer to Strapi's [Creating a new API token documentation](#) for more details and information on regenerating API tokens.
- The **URL**: Use the public URL of your Strapi server, defined in `./config/server.js` as the `url` parameter. Strapi recommends using an absolute URL.
 - For Strapi Cloud projects, use the URL of your Cloud project, for example: `https://my-strapi-project-name.strapiapp.com`

- The **API Version**: Select the version of the API you want your calls to use. Options include:
 - **Version 3**
 - **Version 4**
-

Strava credentials

You can use these credentials to authenticate the following nodes:

- [Strava](#)
- [Strava Trigger](#)

Prerequisites

- Create a [Strava](#) account.
- Create a Strava application in [Settings > API](#). Refer to [Using OAuth2](#) for more information.

Supported authentication methods

- OAuth2

Related resources

Refer to [Strava's API documentation](#) for more information about the service.

Using OAuth2

To configure this credential, you'll need:

- A **Client ID**: Generated when you [create a Strava app](#).
- A **Client Secret**: Generated when you [create a Strava app](#).

Use these settings for your Strava app:

- In n8n, copy the **OAuth Callback URL**. Paste this URL into your Strava app's **Authorization Callback Domain**.
- Remove the protocol (`https://` or `http://`) and the relative URL (`/oauth2/callback` or `/rest/oauth2-credential/callback`) from the **Authorization Callback Domain**. For example, if the OAuth Redirect URL was originally `https://oauth.n8n.cloud/oauth2/callback`, the **Authorization Callback Domain** would be `oauth.n8n.cloud`.
- Copy the **Client ID** and **Client Secret** from your app and add them to your n8n credential.

Refer to [Authentication](#) for more information about Strava's OAuth flow.

Stripe credentials

You can use these credentials to authenticate the following nodes:

- [Stripe Trigger](#)
- [Stripe](#)

Supported authentication methods

- API key

Related resources

Refer to [Stripe's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need a [Stripe](#) admin or developer account and:

- An API **Secret Key**

Before you generate an API key, decide whether to generate it in live mode or test mode. Refer to [Test mode and live mode](#) for more information about the two modes.

Live mode Secret key

To generate a Secret key in live mode:

1. Open the [Stripe developer dashboard](#) and select **API Keys**.
2. In the **Standard Keys** section, select **Create secret key**.
3. Enter a **Key name**, like n8n integration.
4. Select **Create**. The new API key displays.
5. Copy the key and enter it in your n8n credential as the **Secret Key**.

Refer to Stripe's [Create a secret API key](#) for more information.

Test mode Secret key

To use a Secret key in test mode, you must copy the existing one:

1. Go to your [Stripe test mode developer dashboard](#) and select **API Keys**.
2. In the **Standard Keys** section, select **Reveal test key** for the **Secret key**.
3. Copy the key and enter it in your n8n credential as the **Secret Key**.

Refer to Stripe's [Create a secret API key](#) for more information.

Test mode and live mode

All Stripe API requests happen within either [test mode](#) or live mode. Each mode has its own API key.

Use test mode to access simulated test data and live mode to access actual account data. Objects in one mode aren't accessible to the other.

Refer to [API keys | Test mode versus live mode](#) for more information about what's available in each mode and guidance on when to use each.

Key prefixes

Stripes' Secret keys always begin with `sk_`:

- Live keys begin with `sk_live_`.
- Test keys begin with `sk_test_`.

n8n hasn't tested these credentials with Restricted keys (prefixed `rk_`).

Supabase credentials

You can use these credentials to authenticate the following nodes:

- [Supabase](#)
- [Supabase Vector Store](#)

Prerequisites

Create a [Supabase](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Supabase's API documentation](#) for more information about the service.

Using access token

To configure this credential, you'll need:

- A **Host**
- A **Service Role Secret**

To generate your API Key:

1. In your Supabase account, go to the **Dashboard** and create or select a project for which you want to create an API key.
 2. Go to **Project Settings > API** to see the API Settings for your project.
 3. Copy the **URL** from the **Project URL** section and enter it as your n8n **Host**. Refer to [API URL and keys](#) for more detailed instruction.
 4. Reveal and copy the **Project API key** for the `service_role`. Copy that key and enter it as your n8n **Service Role Secret**. Refer to [Understanding API Keys](#) for more information on the `service_role` privileges.
-

SurveyMonkey credentials

You can use these credentials to authenticate the following nodes:

- [SurveyMonkey Trigger](#)

Prerequisites

- Create a [SurveyMonkey](#) account.
- Register an app from your [Developer dashboard > My apps](#).
 - Refer to [Required app scopes](#) for information on the scopes you must use.

Supported authentication methods

- API access token
- OAuth2

Related resources

Refer to [SurveyMonkey's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need:

- An **Access Token**: Generated once you create an app.
- A **Client ID**: Generated once you create an app.
- A **Client Secret**: Generated once you create an app.

Once you've created your app and assigned appropriate scopes, go to **Settings > Credentials**. Copy the **Access Token**, **Client ID**, and **Secret** and add them to n8n.

Using OAuth

To configure this credential, you'll need:

- A **Client ID**: Generated once you create an app.
- A **Client Secret**: Generated once you create an app.

Once you've created your app and assigned appropriate scopes:

1. Go to the app's **Settings > Settings**.
2. From n8n, copy the **OAuth Redirect URL**.
3. Overwrite the app's existing **OAuth Redirect URL** with that URL.
4. Select **Submit Changes**.
5. Be sure the **Scopes** section contains the Required app scopes.

From the app's **Settings > Credentials**, copy the **Client ID** and **Client Secret** and add them to your n8n credential. You can now select **Connect my account** from n8n.

Required app scopes

Once you create your app, go to **Settings > Scopes**. Select these scopes for your n8n credential to work:

- **View Surveys**
- **View Collectors**
- **View Responses**
- **View Response Details**
- **Create/Modify Webhooks**
- **View Webhooks**

Select **Update Scopes** to save them.

SyncroMSP credentials

You can use these credentials to authenticate the following nodes:

- SyncroMSP

Prerequisites

Create a SyncroMSP account.

Supported authentication methods

- API key

Related resources

Refer to SyncroMSP's API documentation for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Called an **API token** in SyncroMSP. To create an API token, go to your **user menu > Profile/Password > API Tokens** and select the option to **Create New Token**. Select **Custom Permissions** to enter a name for your token and adjust the permissions to match your requirements.
- Your **Subdomain**: Enter your SyncroMSP subdomain. This is visible in the URL of your SyncroMSP, located between `https://` and `.syncromsp.com`. If your full URL is `https://n8n-instance.syncromsp.com`, you'd enter `n8n-instance` as the subdomain.

Refer to [API Tokens](#) for more information on creating new tokens.

Sysdig Management credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create a [Sysdig](#) account or configure a local instance.

Supported authentication methods

- Access Key

Related resources

Refer to [Sysdig's documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more.

Using API access key

To configure this credential, you'll need:

- An **Access Key**

Refer to the [Sysdig Agent Access Keys documentation](#) for instructions on obtaining the Access Key from the application.

Taiga credentials

You can use these credentials to authenticate the following nodes:

- [Taiga](#)
- [Taiga Trigger](#)

Prerequisites

Create a [Taiga](#) account.

Supported authentication methods

- Basic auth

Related resources

Refer to [Taiga's API documentation](#) for more information about the service.

Using basic auth

To configure this credential, you'll need:

- A **Username**: Enter your username or user email address. Refer to [Normal login](#) for more information.
- A **Password**: Enter your password.
- The **Environment**: Choose between **Cloud** or **Self-Hosted**. For **Self-Hosted** instances, you'll also need to add:
 - The **URL**: Enter your Taiga URL.

Tapfiliate credentials

You can use these credentials to authenticate the following nodes:

- [Tapfiliate](#)

Prerequisites

Create a [Tapfiliate](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Tapfiliate's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Get your API Key from your [Profile Settings > API](#)

Key

Refer to [Your API key](#) for more information.

Telegram credentials

You can use these credentials to authenticate the following nodes:

- [Telegram](#)
- [Telegram Trigger](#)

Prerequisites

Create a [Telegram](#) account.

Supported authentication methods

- API bot access token

Related resources

Refer to [Telegram's Bot API documentation](#) for more information about the service.

Refer to the [Telegram Bot Features](#) documentation for more information on creating and working with bots.

Using API bot access token

To configure this credential, you'll need:

- A bot **Access Token**

To generate your access token:

1. Start a chat with the [BotFather](#).
2. Enter the /newbot command to create a new bot.
3. The BotFather will ask you for a name and username for your new bot:
 - The **name** is the bot's name displayed in contact details and elsewhere. You can change the bot name later.
 - The **username** is a short name used in search, mentions, and t.me links. Use these guidelines when creating your username:
 - Must be between 5 and 32 characters long.
 - Not case sensitive.
 - May only include Latin characters, numbers, and underscores.
 - Must end in bot, like tetris_bot or TetrisBot.
 - You can't change the username later.
4. Copy the bot **token** the BotFather generates and add it as the **Access Token** in n8n.

Refer to the [BotFather Create a new bot documentation](#) for more information.

TheHive credentials

You can use these credentials to authenticate the following nodes:

- [TheHive](#)

Prerequisites

Install [TheHive](#) on your server.

Supported authentication methods

- API key

Related resources

Refer to [TheHive 3's API documentation](#) and [TheHive 4's API documentation](#) for more information about the services.

Using API key

To configure this credential, you'll need:

- An **API Key**: Create an API key from **Organization > Create API Key**. Refer to [API Authentication](#) for more information.
 - Your **URL**: The URL of your TheHive server.
 - An **API Version**: Choose between:
 - **TheHive 3 (api v0)**
 - **TheHive 4 (api v1)**
 - For TheHive 5, use [TheHive 5 credentials](#) instead.
 - **Ignore SSL Issues**: When turned on, n8n will connect even if SSL certificate validation fails.
-

TheHive 5 credentials

You can use these credentials to authenticate the following nodes with TheHive 5.

- [TheHive 5](#)

Prerequisites

Install [TheHive 5](#) on your server.

Supported authentication methods

- API key

Related resources

Refer to [TheHive's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Users with orgAdmin and superAdmin accounts can generate API keys:
 - orgAdmin account: Go to **Organization > Create API Key** for the user you wish to generate a key for.
 - superAdmin account: Go to **Users > Create API Key** for the user you wish to generate a key for.
 - Refer to [API Authentication](#) for more information.
 - A **URL**: The URL of your TheHive server.
 - **Ignore SSL Issues**: When turned on, n8n will connect even if SSL certificate validation fails.
-

TimescaleDB credentials

You can use these credentials to authenticate the following nodes:

- [TimescaleDB](#)

Prerequisites

An available instance of [TimescaleDB](#).

Supported authentication methods

- Database connection

Related resources

Refer to the [Timescale documentation](#) for more information about the service.

Using database connection

To configure this credential, you'll need:

- The **Host**: The fully qualified server name or IP address of your TimescaleDB server.

- The **Database**: The name of the database to connect to.
- A **User**: The user name you want to log in with.
- A **Password**: Enter the password for the database user you are connecting to.
- **Ignore SSL Issues**: If turned on, n8n will connect even if SSL certificate validation fails and you won't see the **SSL** selector.
- **SSL**: This setting controls the `ssl-mode` connection string for the connection. Options include:
 - **Allow**: Sets the `ssl-mode` parameter to `allow`. First try a non-SSL connection; if that fails, try an SSL connection.
 - **Disable**: Sets the `ssl-mode` parameter to `disable`. Only try a non-SSL connection.
 - **Require**: Sets the `ssl-mode` parameter to `require`, which is the default for TimescaleDB connection strings. Only try an SSL connection. If a root CA file is present, verify that a trusted certificate authority (CA) issued the server certificate.
- **Port**: The port number of the TimescaleDB server.

Refer to the [Timescale connection settings documentation](#) for more information about the non-SSL fields. Refer to [Connect with a stricter SSL](#) for more information about the SSL options.

Todoist credentials

You can use these credentials to authenticate the following nodes:

- [Todoist](#)

Supported authentication methods

- API key
- OAuth2

Related resources

Refer to [Todoist's REST API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need a [Todoist](#) account and:

- An **API Key**

To get your **API Key**:

1. In Todoist, open your [Integration settings](#).
2. Select the **Developer** tab.
3. Copy your **API token** and enter it as the **API Key** in your n8n credential.

Refer to [Find your API token](#) for more information.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you’re [self-hosting](#) n8n, you’ll need a [Todoist](#) account and:

- A **Client ID**
- A **Client Secret**

Get both by creating an application:

1. Open the Todoist [App Management Console](#).
2. Select **Create a new app**.
3. Enter an **App name** for your app, like `n8n integration`.
4. Select **Create app**.
5. Copy the `n8n OAuth Redirect URL` and enter it as the **OAuth redirect URL** in Todoist.
6. Copy the **Client ID** from Todoist and enter it in your n8n credential.
7. Copy the **Client Secret** from Todoist and enter it in your n8n credential.
8. Configure the rest of your Todoist app as it makes sense for your use case.

Refer to the Todoist [Authorization Guide](#) for more information.

Toggl credentials

You can use these credentials to authenticate the following nodes:

- [Toggl Trigger](#)

Prerequisites

Create a [Toggl](#) account.

Supported authentication methods

- Basic auth

Related resources

Refer to [Toggl’s API documentation](#) for more information about the service.

Using basic auth

To configure this credential, you’ll need:

- A **Username**: Enter your user email address.
- A **Password**: Enter your user password.

Refer to [Authentication](#) for more information.

TOTP credentials

You can use these credentials to authenticate the following nodes:

- [TOTP](#)

Prerequisites

Generate a TOTP **Secret** and **Label**.

Supported authentication methods

- Secret and label

Related resources

Time-based One-time Password (TOTP) is an algorithm that generates a one-time password (OTP) using the current time. Refer to [Google Authenticator | Key URI format](#) for more information.

Using secret and label

To configure this credential, you'll need:

- A **Secret**: The secret key encoded in the QR code during authenticator setup. It's an arbitrary key value encoded in Base32, for example: BVDRSBXQB2ZEL5HE. Refer to [Google Authenticator Secret](#) for more information.
 - A **Label**: The identifier for the account. It contains an account name as a URI-encoded string. You can include prefixes to identify the provider or service managing the account. If you use prefixes, use a literal or url-encoded colon to separate the issuer prefix and the account name, for example: GitHub:john-doe. Refer to [Google Authenticator Label](#) for more information.
-

Travis CI credentials

You can use these credentials to authenticate the following nodes:

- [Travis CI](#)

Prerequisites

Create a [Travis CI](#) account.

Supported authentication methods

- API token

Related resources

Refer to [Travis CI's API documentation](#) for more information about the service.

Using API token

To configure this credential, you'll need:

- An **API Token**: Get your API token from [Account Settings > API Token](#) or generate one through the Travis CI [command line client](#).
-

Trellix ePO credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create a [Trellix ePolicy Orchestrator](#) account.

Supported authentication methods

- Basic auth

Related resources

Refer to [Trellix ePO's documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows](#) and [related content](#) on n8n's website.

Using basic auth

To configure this credential, you'll need:

- A **Username** to connect as.
- A **Password** for that user account.

n8n uses These fields to build the -u parameter in the format of -u `username:pw`. Refer to [Web API basics](#) for more information.

Trello credentials

You can use these credentials to authenticate the following nodes:

- [Trello](#)
- [Trello Trigger](#)

Supported authentication methods

- API key

Related resources

Refer to [Trello's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need a [Trello](#) account and:

- An **API Key**
- An **API Token**

To generate both the API Key and API Token, create a Trello Power-Up:

1. Open the Trello [Power-Up Admin Portal](#).
2. Select **New**.
3. Enter a **Name** for your Power-Up, like n8n integration.
4. Select the **Workspace** the Power-Up should have access to.
5. Leave the **iframe connector URL** blank.
6. Enter appropriate contact information.
7. Select **Create**.
8. This should open the Power-Up to the **API Key** page. (If it doesn't, open that page.)
9. Select **Generate a new API Key**.
10. Copy the **API key** from Trello and enter it in your n8n credential.
11. In your Trello API key page, enter your n8n base URL as an **Allowed origin**.
12. In **Capabilities** make sure to select the necessary options.
13. Select the **Token** link next to your Trello **API Key**.
14. When prompted, select **Allow** to grant all the permissions asked for.
15. Copy the Trello **Token** and enter it as the n8n **API Token**.

Refer to Trello's [API Introduction](#) for more information on API keys and tokens. Refer to Trello's [Power-Up Admin Portal](#) for more information on creating Power-Ups.

Twake credentials

You can use these credentials to authenticate the following nodes:

- [Twake](#)

Prerequisites

Create a [Twake](#) account.

Supported authentication methods

- Cloud API key
- Server API key

Related resources

Refer to [Twake's documentation](#) for more information about the service.

Using Cloud API key

To configure this credential, you'll need:

- A **Workspace Key**: Generated when you install the **n8n** application to your Twake Cloud environment and select **Configure**. Refer to [How to connect n8n to Twake](#) for more detailed instructions.

Using Server API key

To configure this credential, you'll need:

- A **Host URL**: The URL of your Twake self-hosted instance.
- A **Public ID**: Generated when you create an app.
- A **Private API Key**: Generated when you create an app.

To generate your **Public ID** and **Private API Key**, [create a Twake application](#):

1. Go to **Workspace Settings > Applications and connectors > Access your applications and connectors > Create an application**.
2. Enter appropriate details.
3. Once you've created your app, view its **API Details**.
4. Copy the **Public identifier** and add it as the n8n **Public ID**.
5. Copy the **Private key** and add it as the n8n **Private API Key**.

Refer to [API settings](#) for more information.

Twilio credentials

You can use these credentials to authenticate the following nodes:

- [Twilio](#)

- [Twilio trigger](#)

Supported authentication methods

- **Auth token:** Twilio recommends this method for local testing only.
- **API key:** Twilio recommends this method for production.

Related resources

Refer to [Twilio's API documentation](#) for more information about the service.

Using Auth Token

To configure this credential, you'll need a [Twilio](#) account and:

- Your Twilio **Account SID**
- Your Twilio **Auth Token**

To set up the credential:

1. In n8n, select **Auth Token** as the **Auth Type**.
2. In Twilio, go to **Console Dashboard > Account Info**.
3. Copy your **Account SID** and enter this in your n8n credential. This acts as a username.
4. Copy your **Auth Token** and enter this in your n8n credential. This acts as a password.

Refer to [Auth Tokens and How to Change Them](#) for more information.

Using API key

To configure this credential, you'll need a [Twilio](#) account and:

- Your Twilio **Account SID**
- An **API Key SID**: Generated when you create an API key.
- An **API Key Secret**: Generated when you create an API key.

To set up the credential:

1. In n8n, select **API Key** as the **Auth Type**.
2. In Twilio, go to **Console Dashboard > Account Info**.
3. Copy your **Account SID** and enter it in your n8n credential.
4. In Twilio, go to your account's **API keys & tokens** page.
5. Select **Create API Key**.
6. Enter a **Friendly name** for your API key, like `n8n integration`.
7. Select your **Key type**. n8n works with either **Main** or **Standard**. Refer to [Selecting an API key type](#) for more information.
8. Select **Create API Key** to finish creating the key.
9. On the **Copy secret key** page, copy the **SID** displayed with the key and enter it in your n8n credential **API Key SID**.
10. On the **Copy secret key** page, copy the **Secret** displayed with the key and enter it in your n8n credential **API Key Secret**.

Refer to [Create an API key](#) for more detailed instructions.

Selecting an API key type

When you create a Twilio API key, you must select a key type. The n8n credential works with **Main** and **Standard** key types.

Here are more details on the different API key types:

- **Main**: This key type gives you the same level of access as using your Account SID and Auth Token in API requests.
- **Standard**: This key type gives you access to all the functionality in Twilio's APIs except the API key resources and Account resources.
- **Restricted**: This key type is in beta. n8n hasn't tested the credential against this key type; if you try it, let us know if you run into any issues.

Refer to [Types of API keys](#) for more information on the key types.

Twist credentials

You can use these credentials to authenticate the following nodes:

- [Twist](#)

Prerequisites

- Create a [Twist](#) account.
- [Create a general integration](#) and configure a valid OAuth Redirect URL. Refer to [Using OAuth2](#) for more information.

Supported authentication methods

- OAuth2

Related resources

Refer to [Twist's API documentation](#) for more information about authenticating with the service.

Using OAuth2

To configure this credential, you'll need:

- A **Client ID**: Generated once you create a general integration.
- A **Client Secret**: Generated once you create a general integration.

To generate your Client ID and Client Secret, [create a general integration](#).

Use these settings for your integration's **OAuth Authentication**:

- Copy the **OAuth Redirect URL** from n8n and enter it as the **OAuth 2 redirect URL** in Twist.

- Select **Update OAuth settings** to save those changes.
- Copy the **Client ID** and **Client Secret** from Twist and enter them in the appropriate fields in n8n.

Local environment redirect URL

Twist doesn't accept a localhost callback URL. These steps should allow you to configure the OAuth credentials for the local environment:

1. Use `ngrok` to expose the local server running on port 5678 to the internet. In your terminal, run the following command:

```
ngrok http 5678
```

2. Run the following command in a new terminal. Replace <YOUR-NGROK-URL> with the URL that you get from the previous step.

```
export WEBHOOK_URL=<YOUR-NGROK-URL>
```

3. Use the generated URL as your **OAuth 2 redirect URL** in Twist.
-

Typeform credentials

You can use these credentials to authenticate the following nodes:

- Typeform Trigger

Supported authentication methods

- API token
- OAuth2

Related resources

Refer to [Typeform's API documentation](#) for more information about the service.

Using API token

To configure this credential, you'll need a Typeform account and:

- A personal **Access Token**

To get your personal access token:

1. Log into your Typeform account.
2. Select your profile avatar in the upper right and go to **Account > Your settings > Personal Tokens**.
3. Select **Generate a new token**.
4. Give your token a **Name**, like n8n integration.
5. For **Scopes**, select **Custom scopes**. Select these scopes:
 - Forms: Read

- Webhooks: Read, Write
6. Select **Generate token**.
 7. Copy the token and enter it in your n8n credential.

Refer to Typeform's [Personal access token documentation](#) for more information.

Using OAuth2

To configure this credential, you'll need a [Typeform](#) account and:

- A **Client ID**: Generated when you register an app.
- A **Client Secret**: Generated when you register an app.

To get your Client ID and Client Secret, register a new Typeform app:

1. Log into your Typeform account.
2. In the upper left, select the dropdown for your organization and select **Developer apps**.
3. Select **Register a new app**.
4. Enter an **App Name** that makes sense, like n8n OAuth2 integration.
5. Enter your n8n base URL as the **App website**, for example <https://n8n-sample.app.n8n.cloud/>.
6. From n8n, copy the **OAuth Redirect URL**. Enter this in Typeform as the **Redirect URI(s)**.
7. Select **Register app**.
8. Copy the **Client Secret** and enter it in your n8n credential.
9. In Typeform, select **Got it** to close the Client Secret modal.
10. The **Developer apps** panel displays your new app. Copy the **Client ID** and enter it in your n8n credential.
11. Once you enter both the **Client ID** and **Client Secret** in n8n, select **Connect my account** and follow the on-screen prompts to finish authorizing the app.

Refer to [Create applications that integrate with Typeform's APIs](#) for more information.

Unleashed Software credentials

You can use these credentials to authenticate the following nodes:

- [Unleashed Software](#)

Prerequisites

Create an [Unleashed Software](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Unleashed's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API ID**: Go to **Integrations > Unleashed API Access** to find your **API ID**.
- An **API Key**: Go to **Integrations > Unleashed API Access** to find your **API Key**.

Refer to [Unleashed API Access](#) for more information.

UpLead credentials

You can use these credentials to authenticate the following nodes:

- [UpLead](#)

Prerequisites

Create an [UpLead](#) account.

Supported authentication methods

- API key

Related resources

Refer to [UpLead's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Go to your **Account > Profiles** to **Generate New API Key**.
-

uProc credentials

You can use these credentials to authenticate the following nodes:

- [uProc](#)

Prerequisites

Create a [uProc](#) account.

Supported authentication methods

- API key

Related resources

Refer to [uProc's API documentation](#) for more information about the service.

Using API Key

To configure this credential, you'll need:

- An **Email** address: Enter the email address you use to log in to uProc. This is also displayed in **Settings > Integrations > API Credentials**.
 - An **API Key**: Go to **Settings > Integrations > API Credentials**. Copy the **API Key (real)** from the **API Credentials** section and enter it in your n8n credential.
-

UptimeRobot credentials

You can use these credentials to authenticate the following nodes:

- [UptimeRobot](#)

Prerequisites

Create an [UptimeRobot](#) account.

Supported authentication methods

- API key

Related resources

Refer to [UptimeRobot's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Get your API Key from **My Settings > API Settings**. Create a **Main API Key** and enter this key in your n8n credential.

API key types

UptimeRobot supports three API key types:

- **Account-specific** (also known as **main**): Pulls data for multiple monitors.
- **Monitor-specific**: Pulls data for a single monitor.
- **Read-only**: Only runs GET API calls.

To complete all of the operations in the UptimeRobot node, use the **Main** or **Account-specific** API key type. Refer to [API authentication](#) for more information.

urlscan.io credentials

You can use these credentials to authenticate the following nodes:

- [urlscan.io](#)

Prerequisites

Create an [urlscan.io](#) account.

Supported authentication methods

- API key

Related resources

Refer to [urlscan.io's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Get your API key from **Settings & API > API Keys**.
-

Venafi TLS Protect Cloud credentials

You can use these credentials to authenticate the following nodes:

- [Venafi TLS Protect Cloud node](#)
- [Venafi TLS Protect Cloud Trigger node](#)

Prerequisites

Create a Venafi [TLS Protect Cloud](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Venafi TLS Protect Cloud's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- A **Region**: Select the region that matches your business needs. Choose **EU** if you're located in the European Union. Otherwise, choose **US**.
 - An **API Key**: Go to your **avatar > Preferences > API Keys** to get your API key. You can also use VCert to get your API key. Refer to [Obtaining an API Key](#) for more information.
-

Venafi TLS Protect Datacenter credentials

You can use these credentials to authenticate the following nodes:

- [Venafi TLS Protect Datacenter node](#)

Prerequisites

- Create a Venafi [TLS Protect Datacenter](#) account.
- Set the expiration and refresh time for tokens. Refer to [Setting up token authentication](#) for more information.
- Create an [API integration](#) in **API > Integrations**. Refer to [Integrating other systems with Venafi products](#) for detailed instructions.
 - Take note of the Client ID for your integration.
 - Choose the scopes needed for the operations you want to perform within n8n. Refer to the scopes table in [Integrating other systems with Venafi products](#) for more details on available scopes.

Supported authentication methods

- API integration

Related resources

Refer to [Venafi's API integration documentation](#) for more information about the service.

Using API integration

To configure this credential, you'll need:

- A **Domain**: Enter your Venafi TLS Protect Datacenter domain.
 - A **Client ID**: Enter the **Client ID** from your API integration. Refer to the information and links in [Prerequisites](#) for more information on creating an API integration.
 - A **Username**: Enter your username.
 - A **Password**: Enter your password.
 - **Allow Self-Signed Certificates**: If turned on, the credential will allow self-signed certificates.
-

Vercel AI Gateway credentials

You can use these credentials to authenticate the following nodes:

- [Chat Vercel AI Gateway](#)

Prerequisites

Create a [Vercel](#) account.

Supported authentication methods

- API key
- OIDC token

Related resources

Refer to the [Vercel AI Gateway documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**

To generate your API Key:

1. [Login to Vercel](#) or [create an account](#).
2. Go to the Vercel dashboard and select the **AI Gateway** tab.
3. Select **API keys** on the left side bar.
4. Select **Add key** and proceed with **Create key** from the Dialog.
5. Copy your key and add it as the **API Key** in n8n.

Using OIDC token

To configure this credential, you'll need:

- An **OIDC token**

To generate your OIDC token:

1. In local development, link your application to a Vercel project with the `vc link` command.
 2. Run the `vercel env pull` command to pull the environment variables from Vercel.
 3. Copy your token and add it as the **OIDC TOKEN** in n8n.
-

Vero credentials

You can use these credentials to authenticate the following nodes:

- [Vero](#)

Prerequisites

Create a [Vero](#) account.

Supported authentication methods

- API auth token

Related resources

Refer to [Vero's API documentation](#) for more information about the service.

Using API auth token

To configure this credential, you'll need:

- An **Auth Token**: Get your auth token from your Vero account [settings](#). Refer to [API authentication](#) for more information.
-

VirusTotal credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create a [VirusTotal](#) account.

Supported authentication methods

- API key

Related resources

Refer to [VirusTotal's documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n's website.

Using API key

To configure this credential, you'll need:

- **An API Token:** Go to your **user account menu > API key** to get your API key. Enter this as the **API Token** in your n8n credential. Refer to [API authentication](#) for more information.
-

Vonage credentials

You can use these credentials to authenticate the following nodes:

- [Vonage](#)

Prerequisites

Create a [Vonage developer](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Vonage's SMS API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**
- An **API Secret**

Get your **API Key** and **API Secret** from your [developer dashboard user account > Settings > API Settings](#). Refer to [Retrieve your account information](#) for more information.

Weaviate credentials

You can use these credentials to authenticate the following nodes:

- [Weaviate Vector Store](#)

Supported authentication methods

- API key

Related resources

Refer to [Weaviate's connection documentation](#) for more information on how to connect to Weaviate.

-8<- “[_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md](#)”

Using API key

Connection type: Weaviate Cloud

Create your [Weaviate Cloud Database](#) and [follow these instructions](#) to get the following parameter values from your Weaviate Cloud Database:

- **Weaviate Cloud Endpoint**
- **Weaviate Api Key**

Note: Weaviate provides a free sandbox option for testing.

Connection type: Custom Connection

For this Connection Type, you need to [deploy Weaviate](#) on your own server, configured so n8n can access it. Refer to [Weaviate's authentication documentation](#) for information on creating and using API keys.

You can then provide the arguments for your custom connection:

- **Weaviate Api Key:** Your Weaviate API key.
- **Custom Connection HTTP Host:** The domain name or IP address of your Weaviate instance to use for HTTP API calls.
- **Custom Connection HTTP Port:** The port your Weaviate instance is running on for HTTP API calls. By default, this is 8080.
- **Custom Connection HTTP Secure:** Whether to connect to the Weaviate through HTTPS for HTTP API calls.
- **Custom Connection gRPC Host:** The hostname or IP address of your Weaviate instance to use for gRPC.
- **Custom Connection gRPC Port:** The gRPC API port for your Weaviate instance. By default, this is 50051.
- **Custom Connection gRPC Secure:** Whether to connect to the Weaviate through HTTPS for gRPC.

For community support, refer to [Weaviate Forums](#).

Webex by Cisco credentials

You can use these credentials to authenticate the following nodes:

- [Webex by Cisco](#)
- [Webex by Cisco Trigger](#)

Prerequisites

Create a [Webex by Cisco](#) account (this should automatically get you [developer account access](#)).

Supported authentication methods

- OAuth2

Related resources

Refer to [Webex's API documentation](#) for more information about the service.

Using OAuth2

Should you need to configure OAuth2 from scratch, you'll need to create an integration to use this credential. Refer to the instructions in the [Webex Registering your Integration documentation](#) to begin.

n8n recommends using the following **Scopes** for your integration:

- spark:rooms_read
- spark:messages_write
- spark:messages_read
- spark:memberships_read
- spark:memberships_write
- meeting:recordings_write
- meeting:recordings_read
- meeting:preferences_read
- meeting:schedules_write
- meeting:schedules_read

Webflow credentials

You can use these credentials to authenticate the following nodes:

- [Webflow](#)
- [Webflow Trigger](#)

Prerequisites

- Create a [Webflow](#) account.
- [Create a site](#): Required for API access token authentication only.

Supported authentication methods

- API access token
- OAuth2

Related resources

Refer to [Webflow's API documentation](#) for more information about the service.

Using API access token

To configure this credential, you'll need:

- A Site **Access Token**: Access tokens are site-specific. Go to your site's **Site Settings > Apps & integrations > API access** and select **Generate API token**. Refer to [Get a Site Token](#) for more information.

Using OAuth2

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you need to configure OAuth2 from scratch, [register an application](#) in your workspace.

Use these settings for your application:

- Copy the **OAuth callback URL** from n8n and add it as a **Redirect URI** in your application.
- Once you've created your application, copy the **Client ID** and **Client Secret** and enter them in your n8n credential.
- If you are using the Webflow Data API V1 (deprecated), enable the **Legacy** toggle. Otherwise, leave this inactive.

Refer to [OAuth](#) for more information on Webflow's OAuth web flow.

Webhook credentials

You can use these credentials to authenticate the following nodes:

- [Webhook](#)

Prerequisites

You must use the authentication method required by the app or service you want to query.

Supported authentication methods

- Basic auth
- Header auth
- JWT auth
- None

-8<- “_snippets/integrations/builtin/credentials/generic-auth/basic-auth.md”

-8<- “_snippets/integrations/builtin/credentials/generic-auth/header-auth.md”

Using JWT auth

JWT Auth is a method of authentication that uses JSON Web Tokens (JWT) to digitally sign data. This authentication method uses the **JWT credential** and can use either a **Passphrase** or **PEM Key** as key type. Refer to [JWT credential](#) for more information.

Wekan credentials

You can use these credentials to authenticate the following nodes:

- [Wekan](#)

Prerequisites

Install [Wekan](#) on your server.

Supported authentication methods

- Basic auth

Related resources

Refer to [Wekan’s API documentation](#) for more information about authenticating with the service.

Using basic auth

To configure this credential, you’ll need:

- A **Username**: Enter your Wekan username.
 - A **Password**: Enter your Wekan password.
 - A **URL**: Enter your Wekan domain.
-

WhatsApp Business Cloud credentials

You can use these credentials to authenticate the following nodes:

- [WhatsApp Business Cloud](#)
- [WhatsApp Trigger](#)

Requirements

To create credentials for WhatsApp, you need the following Meta assets:

- A [Meta developer](#) account: A developer account allows you to create and manage Meta apps, including WhatsApp integrations. ??? note "Set up a Meta developer account"
 1. Visit the [Facebook Developers site](#).
 2. Click **Getting Started** in the upper-right corner (if the link says **My Apps**, you've already set up a developer account).
 3. Agree to terms and conditions.
 4. Provide a phone number for verification.
 5. Select your occupation or role.
- A Meta [business portfolio](#): WhatsApp messaging services require a Meta business portfolio, formerly called a Business Manager account. The UI may still show either option. ??? note "Set up a Meta business portfolio"
 1. Visit the [Facebook Business site](#).
 2. Select **Create an account**.
 - If you already have a Facebook Business account and portfolio, but want a new portfolio, open the business portfolio selector in the left-side menu and select **Create a business portfolio**.
 3. Enter a **Business portfolio name**.
 4. Enter your **name**.
 5. Enter a **business email**.
 6. Select **Submit** or **Create**.
- A Meta [business app](#) configured with WhatsApp: Once you have a developer account, you will create a Meta business app. ??? note "Set up a Meta business app with WhatsApp"
 1. Visit the [Meta for Developers Apps dashboard](#)
 2. Select **Create app**.
 3. In **Add products to your app**, select **Set up** in the WhatsApp tile. Refer to [Add the WhatsApp Product](#) for more detail.
 4. This opens the WhatsApp **Quickstart** page. Select your business portfolio.
 5. Select **Continue**.
 6. In the left-side menu, go to **App settings > Basic**.
 7. Set the **Privacy Policy URL** and **Terms of Service URL** for the app.
 8. Change the **App Mode** to **Live**.

Supported authentication methods

- API key: Use for the [WhatsApp Business Cloud](#) node.
- OAuth2: Use for the [WhatsApp Trigger](#) node.

Related resources

Refer to [WhatsApp's API documentation](#) for more information about the service.

Meta classifies users who create WhatsApp business apps as Tech Providers; refer to Meta's [Get Started for Tech Providers](#) for more information.

Using API key

You need WhatsApp API key credentials to use the [WhatsApp Business Cloud](#) node.

To configure this credential, you'll need:

- An **API Access Token**
- A **Business Account ID**

To generate an access token, follow these steps:

1. Visit the [Meta for Developers Apps dashboard](#).
2. Select your Meta app.
3. In the left-side menu, select **WhatsApp > API Setup**.
4. Select **Generate access token** and confirm the access you want to grant.
5. Copy the **Access token** and add it to n8n as the **Access Token**.
6. Copy the **WhatsApp Business Account ID** and add it to n8n as the **Business Account ID**.

Refer to [Test Business Messaging on WhatsApp](#) for more information on the above steps.

Fully verifying and launching your app will take further configuration. Refer to Meta's [Get Started for Tech Providers](#) Steps 5 and beyond for more information. Refer to [App Review](#) for more information on the Meta App Review process.

Using OAuth2

You need WhatsApp OAuth2 credentials to use the [WhatsApp Trigger](#) node.

To configure this credential, you'll need:

- A **Client ID**
- A **Client Secret**

To retrieve these items, follow these steps:

1. Visit the [Meta for Developers Apps dashboard](#).
2. Select your Meta app.
3. In the left-side menu, select **App settings > Basic**.
4. Copy the **App ID** and enter it as the **Client ID** within the n8n credential.
5. Copy the **App Secret** and enter it as the **Client Secret** within the n8n credential.

Fully verifying and launching your app will take further configuration. Refer to Meta's [Get Started for Tech Providers](#) Steps 5 and beyond for more information. Refer to [App Review](#) for more information on the Meta App Review process.

Wise credentials

You can use these credentials to authenticate the following nodes:

- [Wise](#)
- [Wise Trigger](#)

Prerequisites

Create a [Wise](#) account.

Supported authentication methods

- API token

Related resources

Refer to [Wise's API documentation](#) for more information about the service.

Using API token

To configure this credential, you'll need:

- An **API Token**: Go to your **user menu > Settings > API tokens** to generate an API token. Enter the generated API key in your n8n credential. Refer to [Getting started with the API](#) for more information.
- Your **Environment**: Select the environment that best matches your Wise account environment.
 - If you're using a Wise test sandbox account, select **Test**.
 - Otherwise, select **Live**.
- **Private Key (Optional)**: For live endpoints requiring Strong Customer Authentication (SCA), generate a public and private key. Enter the private key here. Refer to [Add a private key](#) for more information.
 - If you're using a **Test** environment, you'll only need to enter a Private Key if you've enabled Strong Customer Authentication on the [public keys management page](#).

Add a private key

Wise protects some live endpoints and operations with Strong Customer Authentication (SCA). Refer to [Strong Customer Authentication & 2FA](#) for details.

If you make a request to an endpoint that requires SCA, Wise returns a 403 Forbidden HTTP status code. The error returned will look like this:

This request requires Strong Customer Authentication (SCA). Please add a key pair to your account and n8n credentials. See <https://api-docs.transferwise.com/#strong-customer-authentication-personal-token>

To use endpoints requiring SCA, generate an RSA key pair and add the relevant key information to both Wise and n8n:

1. Generate an RSA key pair:

```
$ openssl genrsa -out private.pem 2048  
$ openssl rsa -pubout -in private.pem -out public.pem
```

2. Add the content of the public key `public.pem` to your Wise **user menu > Settings > API tokens > Manage public keys**.
3. Add the content of the private key `private.pem` in n8n to the **Private Key (Optional)**.

Refer to [Personal Token SCA](#) for more information.

Wolfram|Alpha credentials

You can use these credentials to authenticate the following nodes:

- [Wolfram|Alpha](#)

Supported authentication methods

- API key

Related resources

Refer to [Wolfram|Alpha's Simple API documentation](#) for more information about the service.

-8<- “[snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md](#)”

Using API key

To configure this credential, you'll need a registered [Wolfram ID](#) and:

- An **App ID**

To get an App ID:

1. Open the Wolfram|Alpha Developer Portal and go to [API Access](#).
2. Select **Get an App ID**.
3. Enter a **Name** for your application, like `n8n integration`.
4. Enter a **Description** for your application.

5. Select **Simple API** as the **API**.
6. Select **Submit**.
7. Copy the generated **App ID** and enter it in your n8n credential.

Refer to **Getting Started** in the [Wolfram|Alpha Simple API documentation](#) for more information.

Resolve Forbidden connection error

If you enter your App ID and get an error that the credential is **Forbidden**, make sure that you have verified your email address for your Wolfram ID:

1. Go to your [Wolfram ID Details](#).
2. If you don't see the **Verified** label underneath your **Email address**, select the link to **Send a verification email**.
3. You must open the link in that email to verify your email address.

It may take several minutes for the verification to populate to the API, but once it does, retrying the n8n credential should succeed.

WooCommerce credentials

You can use these credentials to authenticate the following nodes:

- [WooCommerce](#)
- [WooCommerce Trigger](#)

Prerequisites

- Install the [WooCommerce](#) plugin on your WordPress website.
- In WordPress, go to **Settings > Permalinks** and set your WordPress permalinks to use something other than **Plain**.

Supported authentication methods

- API key

Related resources

Refer to [WooCommerce's REST API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- A **Consumer Key**: Created when you generate an API key.
- A **Consumer Secret**: Created when you generate an API key.
- A **WooCommerce URL**

To generate an API key and set up your credential:

1. Go to **WooCommerce > Settings > Advanced > Rest API > Add key**.
2. Select **Read/Write** from the **Permissions** dropdown.
3. Copy the generated **Consumer Key** and **Consumer Secret** and enter them into your n8n credentials.
4. Enter your WordPress site URL as the **WooCommerce URL**.
5. By default, n8n passes your credential details in the Authorization header. If you need to pass them as query string parameters instead, turn on **Include Credentials in Query**.

Refer to [Generate Keys](#) for more information.

Resolve “Consumer key is missing” error

When you try to connect your credentials, you may receive an error like this: Consumer key is missing.

This occurs when the server can't parse the Authorization header details when authenticating over SSL.

To resolve it, turn on the **Include Credentials in Query** toggle to pass the consumer key/secret as query string parameters instead and retry the credential.

WordPress credentials

You can use these credentials to authenticate the following nodes:

- [WordPress](#)

Prerequisites

- Create a [WordPress](#) account or deploy WordPress on a server.

Supported authentication methods

- Basic auth

Related resources

Refer to [WordPress's API documentation](#) for more information about the service.

Using basic auth

To configure this credential, you'll need:

- Your WordPress **Username**
- A WordPress application **Password**

- Your **WordPress URL**
- Decide whether to **Ignore SSL Issues**

Using this credential involves three steps:

1. Enable two-step authentication.
2. Create an application password.
3. Set up the credential.

Refer to the detailed instructions below for each step.

Enable two-step authentication

To generate an application password, you must first enable Two-Step Authentication in WordPress. If you've already done this, skip to the next section.

1. Open your WordPress profile.
2. Select **Security** from the left menu.
3. Select **Two-Step Authentication**. The **Two-Step Authentication** page opens.
4. If Two-Step Authentication isn't enabled, you must enable it.
5. Choose whether to enable it using an authenticator app or SMS codes and follow the on-screen instructions.

Refer to WordPress's Enable Two-Step Authentication for detailed instructions.

Create an application password

With Two-Step Authentication enabled, you can now generate an application password:

1. From the WordPress **Security > Two-Step Authentication** page, select **+ Add new application password** in the **Application passwords** section.
2. Enter an **Application name**, like n8n integration.
3. Select **Generate Password**.
4. Copy the password it generates. You'll use this in your n8n credential.

Set up the credential

Congratulations! You're now ready to set up your n8n credential:

1. Enter your WordPress **Username** in your n8n credential.
2. Enter the application password you copied above as the **Password** in your n8n credential.
3. Enter the URL of your WordPress site as the **WordPress URL**.
4. Optional: Use the **Ignore SSL Issues** to choose whether you want the n8n credential to connect even if SSL certificate validation fails (turned on) or whether to respect SSL certificate validation (turned off).

Workable credentials

You can use these credentials to authenticate the following nodes:

- [Workable Trigger](#)

Prerequisites

Create a [Workable](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Workable's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- A **Subdomain**: Your Workable subdomain is the part of your Workable domain between `https://` and `.workable.com`. So if the full domain is `https://n8n.workable.com`, the subdomain is `n8n`. The subdomain is also displayed on your Workable **Company Profile** page.
 - An **Access Token**: Go to your **profile > Integrations > Apps** and select **Generate API token**. Refer to [Generate a new token](#) for more information.
-

Wufoo credentials

You can use these credentials to authenticate the following nodes:

- [Wufoo Trigger](#)

Prerequisites

Create a [Wufoo](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Wufoo's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: Get your API key from the [Wufoo Form Manager](#). To the right of a form, select **More > API Information**. Refer to [Using API Information and Webhooks](#) for more information.
 - A **Subdomain**: Your subdomain is the part of your Wufoo URL that comes after `https://` and before `wufoo.com`. So if the full domain is `https://n8n.wufoo.com`, the subdomain is `n8n`. Admins can view the subdomain in the [Account Manager](#). Refer to [Your Subdomain](#) for more information.
-

X (formerly Twitter) credentials

You can use these credentials to authenticate the following nodes:

- [X \(formerly Twitter\)](#)

Prerequisites

- Create an [X developer account](#).
- Create a [Twitter app](#) or use the default project and app created when you sign up for the developer portal. Refer to each supported authentication method below for more details on the app's configuration.

Supported authentication methods

- OAuth2

Related resources

Refer to [X's API documentation](#) for more information about the service. Refer to [X's API authentication documentation](#) for more information about authenticating with the service.

Refer to [Application-only Authentication](#) for more information about app-only authentication.

Using OAuth2

Use this method if you're using n8n version 0.236.0 or later.

To configure this credential, you'll need:

- A **Client ID**
- A **Client Secret**

To generate your Client ID and Client Secret:

1. In the Twitter [developer portal](#), open your project.

2. On the project's **Overview** tab, find the **Apps** section and select **Add App**.
3. Give your app a **Name** and select **Next**.
4. Go to the **App Settings**.
5. In the **User authentication settings**, select **Set Up**.
6. Set the **App permissions**. Choose **Read and write and Direct message** if you want to use all functions of the n8n X node.
7. In the **Type of app** section, select **Web App, Automated App or Bot**.
8. In n8n, copy the **OAuth Redirect URL**.
9. In your X app, find the **App Info** section and paste that URL in as the **Callback URI / Redirect URL**.
10. Add a **Website URL**.
11. Save your changes.
12. Copy the **Client ID** and **Client Secret** displayed in X and add them to the corresponding fields in your n8n credential.

Refer to X's [OAuth 2.0 Authentication documentation](#) for more information on working with this authentication method.

X rate limits

X has time-based rate limits per endpoint based on your developer access plan level. X calculates app rate limits and user rate limits independently. Refer to [Rate limits](#) for the access plan level rate limits and guidance on avoiding hitting them.

Use the guidance below for calculating rate limits:

- If you're using the deprecated OAuth method, user rate limits apply. You'll have one limit per time window for each set of users' access tokens.
- If you're [Using OAuth2](#), app rate limits apply. You'll have a limit per time window for requests made by your app.

X calculates user rate limits and app rate limits independently.

Refer to X's [Rate limits and authentication methods](#) for more information about these rate limit types.

xAI credentials

You can use these credentials to authenticate the following nodes:

- [Chat xAI Grok](#)

Prerequisites

Create an [xAI](#) account.

Supported authentication methods

- API key

Related resources

Refer to [xAI's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- An **API Key**: You can create a new API key on the [xAI Console API Keys page](#).

Refer to the [The Hitchhiker's Guide to Grok | xAI](#) for more information.

Xata credentials

You can use these credentials to authenticate the following nodes:

- [Xata](#)

Prerequisites

Create a [Xata](#) database or an account on an existing database.

Supported authentication methods

- API key

Related resources

Refer to [Xata's documentation](#) for more information about the service.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Using API key

To configure this credential, you'll need:

- The **Database Endpoint**: The Workspace API requires that you identify the database you're requesting information from using this format: `https://{{workspace-display-name}}-{{workspace-id}}.{{region}}.xata.sh/db/{{dbname}}`. Refer to [Workspace API](#) for more information.
 - `{{workspace-display-name}}`: The workspace display name is an optional identifier you can include in your Database Endpoint. The API ignores it, but including it can make it easier to figure out which workspace this database is in if you're saving multiple credentials.
 - `{{workspace-id}}`: The unique ID of the workspace,

- alphanumeric characters.
- {region}: The hosting region for the database. This value must match the database region configuration.
 - {dbname}: The name of the database you're interacting with.
- A **Branch**: Enter the name of the GitHub branch for your database.
 - An **API Key**: To generate an API key, go to [Account Settings](#) and select **+ Add a key**. Refer to [Generate an API Key](#) for more information.
-

Xero credentials

You can use these credentials to authenticate the following nodes:

- [Xero](#)

Prerequisites

Create a [Xero](#) account.

Supported authentication methods

- OAuth2

Related resources

Refer to [Zero's API documentation](#) for more information about the service.

Using OAuth2

To configure this credential, you'll need:

- A **Client ID**: Generated when you create a new app for a custom connection.
- A **Client Secret**: Generated when you create a new app for a custom connection.

To generate your Client ID and Client Secret, [create an OAuth2 custom connection app](#) in your Xero developer portal [My Apps](#).

Use these settings for your app:

- Select **Web app** as the **Integration Type**.
- For the **Company or Application URL**, enter the URL of your n8n server or reverse proxy address. For cloud users, for example, this is: `https://your-username.app.n8n.cloud/`.
- Copy the **OAuth Redirect URL** from n8n and add it as an **OAuth 2.0 redirect URI** in your app.
- Select appropriate **scopes** for your app. Refer to [OAuth2 Scopes](#) for more information.
 - To use all functionality in the [Xero](#) node, add the

accounting.contacts and accounting.transactions scopes.

Refer to Xero's [OAuth Custom Connections](#) documentation for more information.

Yourls credentials

You can use these credentials to authenticate the following nodes:

- [Yourls](#)

Prerequisites

Install [Yourls](#) on your server.

Supported authentication methods

- API key

Related resources

Refer to [Yourl's documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- A **Signature** token: Go to **Tools > Secure passwordless API call** to get your **Signature** token. Refer to [Yourl's Passworldess API documentation](#) for more information.
 - A **URL**: Enter the URL of your Yourls instance.
-

Zabbix credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create a [Zabbix Cloud](#) account or self-host your own Zabbix server.

Supported authentication methods

- API key

Related resources

Refer to [Zabbix's API documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n's website.

Using API key

To configure this credential, you'll need:

- an **API Token**: An API key for your Zabbix user.
- the **URL**: The URL of your Zabbix server. Don't include /zabbix as part of the URL.

Refer to [Zabbix's API documentation](#) for more information about authenticating to the service.

Zammad credentials

You can use these credentials to authenticate the following nodes:

- [Zammad](#)

Prerequisites

- Create a hosted [Zammad](#) account or set up your own Zammad instance.
- For token authentication, enable **API Token Access** in **Settings > System > API**. Refer to [Setting up a Zammad](#) for more information.

Supported authentication methods

- Basic auth
- Token auth: Zammad recommends using this authentication method.

Related resources

Refer to [Zammad's API Authentication documentation](#) for more information about authenticating with the service.

Using basic auth

To configure this credential, you'll need:

- A **Base URL**: Enter the URL of your Zammad instance.
- An **Email** address: Enter the email address you use to log in to Zammad.

- **A Password:** Enter your Zammad password.
- **Ignore SSL Issues:** When turned on, n8n will connect even if SSL certificate validation fails.

Using token auth

To configure this credential, you'll need:

- A **Base URL:** Enter the URL of your Zammad instance.
- An **Access Token:** Once **API Token Access** is enabled for the Zammad instance, any user with the `user_preferences.access_token` permission can generate an **Access Token** by going to your **avatar > Profile > Token Access** and **Create** a new token.
 - The access token permissions depend on what actions you'd like to complete with this credential. For all functionality within the Zammad node, select:
 - `admin.group`
 - `admin.organization`
 - `admin.user`
 - `ticket.agent`
 - `ticket.customer`
- **Ignore SSL Issues:** When turned on, n8n will connect even if SSL certificate validation fails.

Zendesk credentials

You can use these credentials to authenticate the following nodes:

- Zendesk
- Zendesk Trigger

Prerequisites

- Create a Zendesk account.
- For API token authentication, enable token access to the API in Admin Center under **Apps and integrations > APIs > Zendesk APIs**.

Supported authentication methods

- API token
- OAuth2

Related resources

Refer to Zendesk's API documentation for more information about the service.

Using API token

To configure this credential, you'll need:

- Your **Subdomain**: Your Zendesk subdomain is the portion of the URL between `https://` and `.zendesk.com`. For example, if the Zendesk URL is `https://n8n-example.zendesk.com/agent/dashboard`, the subdomain is `n8n-example`.
- An **Email** address: Enter the email address you use to log in to Zendesk.
- An **API Token**: Generate an API token in **Apps and integrations > APIs > Zendesk API**. Refer to [API token](#) for more information.

Using OAuth2

To configure this credential, you'll need:

- A **Client ID**: Generated when you create a new OAuth client.
- A **Client Secret**: Generated when you create a new OAuth client.
- Your **Subdomain**: Your Zendesk subdomain is the portion of the URL between `https://` and `.zendesk.com`. For example, if the Zendesk URL is `https://n8n-example.zendesk.com/agent/dashboard`, the subdomain is `n8n-example`.

To create a new OAuth client, go to **Apps and integrations > APIs > Zendesk API > OAuth Clients**.

Use these settings:

- Copy the **OAuth Redirect URL** from n8n and enter it as a **Redirect URL** in the OAuth client.
- Copy the **Unique identifier** for the Zendesk client and enter this as your n8n **Client ID**.
- Copy the **Secret** from Zendesk and enter this as your n8n **Client Secret**

Refer to [Registering your application with Zendesk](#) for more information.

Zep credentials

You can use these credentials to authenticate the following nodes:

- [Zep](#)
- [Zep Vector Store](#)

Supported authentication methods

- API key

Related resources

Refer to [Zep's Cloud SDK documentation](#) for more information about the service. Refer to [Zep's REST API documentation](#) for information about the API.

-8<- “ snippets/integrations/builtin/cluster-nodes/langchain-overview-link.md”

Using API key

To configure this credential, you'll need a [Zep server](#) with at least one project and:

- An **API URL**
- An **API Key**

Setup depends on whether you're using Zep Cloud or self-hosted Zep Open Source.

Zep Cloud setup

Follow these instructions if you're using [Zep Cloud](#):

1. In Zep, open the **Project Settings**.
2. In the **Project Keys** section, select **Add Key**.
3. Enter a **Key Name**, like n8n integration.
4. Select **Create**.
5. Copy the key and enter it in your n8n integration as the **API Key**.
6. Turn on the **Cloud** toggle.

Self-hosted Zep Open Source setup

Follow these instructions if you're self-hosting Zep Open Source:

1. Enter the JWT token for your Zep server as the **API Key** in n8n.
 2. Make sure the **Cloud** toggle is off.
 3. Enter the URL for your Zep server as the **API URL**.
-

Zoho credentials

You can use these credentials to authenticate the following nodes:

- [Zoho CRM](#)

Prerequisites

Create a [Zoho](#) account.

Supported authentication methods

- OAuth2

Related resources

Refer to [Zoho's CRM API documentation](#) for more information about the service.

Using OAuth2

To configure this credential, you'll need:

- An **Access Token URL**: Zoho provides region-specific access token URLs. Select the region that best fits your Zoho data center:
 - **AU**: Select this option for Australia data center.
 - **CN**: Select this option for Canada data center.
 - **EU**: Select this option for the European Union data center.
 - **IN**: Select this option for the India data center.
 - **US**: Select this option for the United States data center.

Refer to [Multi DC](#) for more information about selecting a data center.

-8<- “_snippets/integrations/builtin/credentials/cloud-oauth-button.md”

If you need to configure OAuth2 from scratch, [register an application](#) with Zoho.

Use these settings for your application:

- Select **Server-based Applications** as the **Client Type**.
 - Copy the **OAuth Callback URL** from n8n and enter it in the Zoho **Authorized Redirect URIs** field.
 - Copy the **Client ID** and **Client Secret** from the application and enter them in your n8n credential.
-

Zoom credentials

You can use these credentials to authenticate the following nodes:

- [Zoom](#)

Prerequisites

Create a [Zoom](#) account. Your account must have one of the following permissions:

- Account owner
- Account admin
- Zoom for developers role

Supported authentication methods

- API JWT token
- OAuth2

Related resources

Refer to [Zoom's API documentation](#) for more information about the service.

Using API JWT token

This authentication method has been fully deprecated by Zoom. Don't create new credentials with it.

To configure this credential, you'll need:

- A **JWT token**: To create a JWT token, create a new JWT app in the [Zoom App Marketplace](#).

Using OAuth2

To configure this credential, you'll need:

- A **Client ID**: Generated when you create an OAuth app on the Zoom App Marketplace.
- A **Client Secret**: Generated when you create an OAuth app.

To generate your **Client ID** and **Client Secret**, [create an OAuth app](#).

Use these settings for your OAuth app:

- Select **User-managed app** for **Select how the app is managed**.
 - Copy the **OAuth Callback URL** from n8n and enter it as an **OAuth Redirect URL** in Zoom.
 - If your n8n credential displays a **Whitelist URL**, also enter that URL as a **OAuth Redirect URL**.
 - Enter **Scopes** for the scopes you plan to use. For all functionality in the [Zoom](#) node, select:
 - meeting:read
 - meeting:write
 - Refer to [OAuth scopes | Meeting scopes](#) for more information on meeting scopes.
 - Copy the **Client ID** and **Client Secret** provided in the Zoom app and enter them in your n8n credential.
-

Zscaler ZIA credentials

-8<- “_snippets/integrations/builtin/credentials/cred-only-statement.md”

Prerequisites

Create an admin account on a [Zscaler Internet Access \(ZIA\)](#) cloud instance.

Supported authentication methods

- Basic auth and API key combo

Related resources

Refer to [Zscaler ZIA's documentation](#) for more information about the service.

This is a credential-only node. Refer to [Custom API operations](#) to learn more. View [example workflows and related content](#) on n8n's website.

Using basic auth and API key combo

To configure this credential, you'll need:

- A **Base URL**: Enter the base URL of your Zscaler ZIA cloud name. To get your base URL, log in to the ZIA Admin Portal and go to **Administration > Cloud Service API Security**. The base URL is displayed in both the **Cloud Service API Key** tab and the **OAuth 2.0 Authorization Servers** tab.
- A **Username**: Enter your ZIA admin username.
- A **Password**: Enter your ZIA admin password.
- An **Api Key**: Get an API key by creating one from **Administration > Cloud Service API Security > Cloud Service API Key**.

Refer to [About Cloud Service API Key](#) for more detailed instructions.

Zulip credentials

You can use these credentials to authenticate the following nodes:

- [Zulip](#)

Prerequisites

Create a [Zulip](#) account.

Supported authentication methods

- API key

Related resources

Refer to [Zulip's API documentation](#) for more information about the service.

Using API key

To configure this credential, you'll need:

- A **URL**: Enter the URL of your Zulip domain.
 - An **Email** address: Enter the email address you use to log in to Zulip.
 - An **API Key**: Get your API key in the **Gear cog > Personal Settings > Account & privacy > API Key**. Refer to [API Keys](#) for more information.
-

Custom API operations

-8<- “_snippets/integrations/credential-only-intro.md”

Predefined credential types

A predefined credential type is a credential that already exists in n8n. You can use predefined credential types instead of generic credentials in the HTTP Request node.

For example: you create an Asana credential, for use with the Asana node. Later, you want to perform an operation that isn't supported by the Asana node, using Asana's API. You can use your existing Asana credential in the HTTP Request node to perform the operation, without additional authentication setup.

Using predefined credential types

-8<- “_snippets/integrations/predefined-credential-type-how-to.md”

Credential scopes

Some existing credential types have specific scopes: endpoints that they work with. n8n warns you about this when you select the credential type.

For example, follow the steps in [Using predefined credential types](#), and select **Google Calendar OAuth2 API** as your **Credential Type**. n8n displays a box listing the two endpoints you can use this credential type with:

[Image: The scopes box]

Handling API rate limits

API rate limits are restrictions on request frequency. For example, an API may limit the number of requests you can make per minute, or per day.

APIs can also limit how much data you can send in one request, or how much data the API sends in a single response.

Identify rate limit issues

When an n8n node hits a rate limit, it errors. n8n displays the error message in the node output panel. This includes the error message from the service.

If n8n received error 429 (too many requests) from the service, the error message is **The service is receiving too many requests from you**.

To check the rate limits for the service you're using, refer to the API documentation for the service.

Handle rate limits for integrations

There are two ways to handle rate limits in n8n's integrations: using the Retry On Fail setting, or using a combination of the [Loop Over Items](#) and [Wait](#) nodes:

- Retry On Fail adds a pause between API request attempts.
- With Loop Over Items and Wait you can break your request data into smaller chunks, as well as pausing between requests.

Enable Retry On Fail

When you enable Retry On Fail, the node automatically tries the request again if it fails the first time.

1. Open the node.
2. Select **Settings**.
3. Enable the **Retry On Fail** toggle.
4. Configure the retry settings: if using this to work around rate limits, set **Wait Between Tries (ms)** to more than the rate limit. For example, if the API you're using allows one request per second, set **Wait Between Tries (ms)** to 1000 to allow a 1 second wait.

Use Loop Over Items and Wait

Use the Loop Over Items node to batch the input items, and the Wait node to introduce a pause between each request.

1. Add the Loop Over Items node before the node that calls the API. Refer to [Loop Over Items](#) for information on how to configure the node.
2. Add the Wait node after the node that calls the API, and connect it back to the Loop Over Items node. Refer to [Wait](#) for information on how to configure the node.

For example, to handle rate limits when using OpenAI:

[Image: "Screenshot of a workflow using the Loop Over Items node and Wait node to handle API rate limits for the OpenAI APIs"]

Handle rate limits in the HTTP Request node

The HTTP Request node has built-in settings for handling rate limits and large amounts of data.

Batch requests

Use the Batching option to send more than one request, reducing the request size, and introducing a pause between requests. This is the equivalent of using Loop Over Items and Wait.

1. In the HTTP Request node, select **Add Option > Batching**.
2. Set **Items per Batch**: this is the number of input items to include in each request.
3. Set **Batch Interval (ms)** to introduce a delay between requests.

For example, if the API you're using allows one request per second, set **Wait Between Tries (ms)** to 1000 to allow a 1 second wait.

Paginate results

APIs paginate their results when they need to send more data than they can handle in a single response. For more information on pagination in the HTTP Request node, refer to [HTTP Request node | Pagination](#).

Install and manage community nodes

There are three ways to install community nodes:

- Within n8n using the [nodes panel](#) (for verified community nodes only).
 - Within n8n [using the GUI](#): Use this method to install community nodes from the npm registry.
 - [Manually from the command line](#): use this method to install community nodes from npm if your n8n instance doesn't support installation through the in-app GUI.
-

Install verified community nodes in the n8n app

Install a community node

To install a [verified community node](#):

1. Go to the **Canvas** and open the **nodes panel** (either by selecting '+' or pressing ++tab++).
2. **Search** for the node that you're looking for. If there is a matching verified community node, you will see a **More from the community** section at the bottom of the nodes panel.
3. Select the node you want to install. This takes you to a detailed view of the node, showing all the supported actions.
4. Select **install**. This will install the node for your instance and enable all members to use it in their workflows.
5. You can now add the node to your workflows.

Uninstall a community node

To uninstall a community node:

1. Go to **Settings > Community nodes**.
2. On the node you want to install, select **Options** Image: Three dots options menu.
3. Select **Uninstall package**.

4. Select **Uninstall Package** in the confirmation modal.
-

Install community nodes from npm in the n8n app

Install a community node

To install a community node from npm:

1. Go to **Settings > Community Nodes**.
2. Select **Install**.
3. Find the node you want to install:
 1. Select **Browse**. n8n takes you to an npm search results page, showing all npm packages tagged with the keyword `n8n-community-node-package`.
 2. Browse the list of results. You can filter the results or add more keywords.
 3. Once you find the package you want, make a note of the package name. If you want to install a specific version, make a note of the version number as well.
 4. Return to n8n.
4. Enter the npm package name, and version number if required. For example, consider a community node designed to access a weather API called "Storms." The package name is `n8n-node-storms`, and it has three major versions.
 - To install the latest version of a package called `n8n-node-weather`: enter `n8n-nodes-storms` in **Enter npm package name**.
 - To install version 2.3: enter `n8n-node-storms@2.3` in **Enter npm package name**.
5. Agree to the risks of using community nodes: select **I understand the risks of installing unverified code from a public source**.
6. Select **Install**. n8n installs the node, and returns to the **Community Nodes** list in **Settings**.

Uninstall a community node

To uninstall a community node:

1. Go to **Settings > Community nodes**.
2. On the node you want to install, select **Options** Image: Three dots options menu.
3. Select **Uninstall package**.
4. Select **Uninstall Package** in the confirmation modal.

Upgrade a community node

Upgrade to the latest version

You can upgrade community nodes to the latest version from the node list in **Settings > community nodes**.

When a new version of a community node is available, n8n displays an **Update** button on the node. Click the button to upgrade to the latest version.

Upgrade to a specific version

To upgrade to a specific version (a version other than the latest), uninstall the node, then reinstall it, making sure to specify the target version. Follow the [Installation](#) instructions for more guidance.

Downgrade a community node

If there is a problem with a particular version of a community node, you may want to roll back to a previous version.

To do this, uninstall the community node, then reinstall it, targeting a specific node version. Follow the [Installation](#) instructions for more guidance.

Manually install community nodes from npm

You can manually install community nodes from the npm registry on self-hosted n8n.

You need to manually install community nodes in the following circumstances:

- Your n8n instance runs in queue mode.
- You want to install [private packages](#).

Install a community node

Access your Docker shell:

```
docker exec -it n8n sh
```

Create `~/.n8n/nodes` if it doesn't already exist, and navigate into it:

```
mkdir ~/.n8n/nodes  
cd ~/.n8n/nodes
```

Install the node:

```
npm i n8n-nodes-nodeName
```

Then restart n8n.

Uninstall a community node

Access your Docker shell:

```
docker exec -it n8n sh
```

Run npm uninstall:

```
npm uninstall n8n-nodes-nodeName
```

Upgrade a community node

Upgrade to the latest version

Access your Docker shell:

```
docker exec -it n8n sh
```

Run npm update:

```
npm update n8n-nodes-nodeName
```

Upgrade or downgrade to a specific version

Access your Docker shell:

```
docker exec -it n8n sh
```

Run npm uninstall to remove the current version:

```
npm uninstall n8n-nodes-nodeName
```

Run npm install with the version specified:

```
# Replace 2.1.0 with your version number  
npm install n8n-nodes-nodeName@2.1.0
```

Risks when using community nodes

Installing community nodes from npm means you are installing unverified code from a public source into your n8n instance. This has some risks.

Risks include:

- System security: community nodes have full access to the machine that n8n runs on, and can do anything, including malicious actions.
- Data security: any community node that you use has access to data in your workflows.
- Breaking changes: node developers may introduce breaking changes in new versions of their nodes. A breaking change is an update that breaks previous functionality. Depending on the node versioning approach that a node developer chooses, upgrading to a version with a breaking change could cause all workflows using the node to break. Be careful when upgrading your nodes.

Report bad community nodes

You can report bad community nodes to security@n8n.io

Disable community nodes

If you are self-hosting n8n, you can disable community nodes by setting `N8N_COMMUNITY_PACKAGES_ENABLED` to false. On n8n cloud, visit the [Cloud Admin Panel](#) and disable community nodes from there. See [troubleshooting](#) for more information.

n8n community node blocklist

n8n maintains a blocklist of community nodes. You can't install any node on this list.

n8n may add community nodes to the blocklist for a range of reasons, including:

- The node is intentionally malicious
- It's low quality (low enough to be harmful)

If you are a community node creator whose node is on the blocklist, and you believe this is a mistake, contact hello@n8n.io.

Using community nodes

To use community nodes, you first need to [install](#) them.

Adding community nodes to your workflow

After installing a community node, you can use it like any other node. n8n displays the node in search results in the **Nodes** panel. n8n marks community nodes with a **Package** Image: Package icon icon in the nodes panel.

Community nodes with duplicate names

It's possible for several community nodes to have the same name. If you use two nodes with the same name in your workflow, they'll look the same, unless they have different icons.

Troubleshooting and errors

Error: Missing packages

n8n installs community nodes directly onto the hard disk. The files must be available at startup for n8n to load them. If the packages aren't available at startup, you get an error warning of missing packages.

If running n8n using Docker: depending on your Docker setup, you may lose the packages when you recreate your container or upgrade your n8n version. You must either:

- Persist the contents of the `~/.n8n/nodes` directory. This is the best option. If you follow the [Docker installation guide](#), the setup steps include persisting this directory.
- Set the `N8N_REINSTALL_MISSING_PACKAGES` environment variable to `true`.

The second option might increase startup time and may cause health checks to fail.

Prevent loading community nodes on n8n cloud

If your n8n cloud instance crashes and fails to start, you can prevent installed community nodes from loading on instance startup. Visit the [Cloud Admin Panel](#) > **Manage** and toggle **Disable all community nodes** to `true`. This toggle is only visible when you allow community node installation.

Building community nodes

-8<- “_snippets/integrations/submit-community-node.md”

Creating nodes

Learn how to build your own custom [nodes](#).

This section includes:

- Guidance on planning your build, including [which style to use](#).
- [Tutorials](#) for different node building styles.
- Instructions for [testing your node](#), including how to use the n8n [node linter](#) and [troubleshooting support](#).
- How to [share your node](#) with the community, submit it for [verification by n8n](#), or use it as a [private node](#).
- [Reference material](#), including UI elements and information on the individual files that make up a node.

Prerequisites

This section assumes the following:

- Some familiarity with JavaScript and TypeScript.
- Ability to manage your own development environment, including

git.

- Knowledge of npm, including creating and submitting packages.
 - Familiarity with n8n, including a good understanding of data structures and item linking.
-

Plan a node

This section provides guidance on designing your node, including key technical decisions such as choosing your node building style.

When building a node, there are design choices you need to make before you start:

- Which node type you need to build.
 - Which node building style to use.
 - Your UI design and UX principles
 - Your node's file structure.
-

Node types: Trigger and Action

There are two node types you can build for n8n: trigger nodes and action nodes.

Both types provide integrations with external services.

Trigger nodes

Trigger nodes start a workflow and supply the initial data. A workflow can contain multiple trigger nodes but with each execution, only one of them will execute, depending on the triggering event.

There are three types of trigger nodes in n8n:

| Type | Description | Example Nodes |
|---------|---|--|
| Webhook | Nodes for services that support webhooks. These nodes listen for events and trigger workflows in real time. | Zendesk Trigger ,
Telegram Trigger ,
Brevo Trigger |
| Polling | Nodes for services that don't support webhooks. These nodes periodically check for new data, triggering workflows when they detect updates. | Airtable Trigger ,
Gmail Trigger ,
Google Sheet Trigger ,
RssFeed Trigger ,
Read Trigger |
| | Nodes that handle real-time responses not related to HTTP requests or polling. | AMQP Trigger ,
RabbitMQ Trigger , |

| | | |
|--------|--|--|
| Others | This includes message queue nodes and time-based triggers. | MQTT Trigger ,
Schedule Trigger ,
Email Trigger (IMAP) |
|--------|--|--|

Action nodes

Action nodes perform operations as part of your workflow. These can include manipulating data, and triggering events in other systems.

Choose your node building approach

n8n has two node-building styles, declarative and programmatic.

You should use the declarative style for most nodes. This style:

- Uses a JSON-based syntax, making it simpler to write, with less risk of introducing bugs.
- Is more future-proof.
- Supports integration with REST APIs.

The programmatic style is more verbose. You must use the programmatic style for:

- Trigger nodes
- Any node that isn't REST-based. This includes nodes that need to call a GraphQL API and nodes that use external dependencies.
- Any node that needs to transform incoming data.
- Full versioning. Refer to [Node versioning](#) for more information on types of versioning.

Data handling differences

The main difference between the declarative and programmatic styles is how they handle incoming data and build API requests. The programmatic style requires an `execute()` method, which reads incoming data and parameters, then builds a request. The declarative style handles this using the `routing` key in the `operations` object. Refer to [Node base file](#) for more information on node parameters and the `execute()` method.

Syntax differences

To understand the difference between the declarative and programmatic styles, compare the two code snippets below. This example creates a simplified version of the SendGrid integration, called "FriendGrid." The following code snippets aren't complete: they emphasize the differences in the node building styles.

In programmatic style:

```
import {
```

```

    IExecuteFunctions,
    INodeExecutionData,
    INodeType,
    INodeTypeDescription,
    IRequestOptions,
} from 'n8n-workflow';

// Create the FriendGrid class
export class FriendGrid implements INodeType {
    description: INodeTypeDescription = {
        displayName: 'FriendGrid',
        name: 'friendGrid',
        ...
        properties: [
            {
                displayName: 'Resource',
                ...
            },
            {
                displayName: 'Operation',
                name: 'operation',
                type: 'options',
                displayOptions: {
                    show: [
                        resource: [
                            'contact',
                        ],
                    ],
                },
                options: [
                    {
                        name: 'Create',
                        value: 'create',
                        description: 'Create a contact',
                    },
                ],
                default: 'create',
                description: 'The operation to perform.',
            },
            {
                displayName: 'Email',
                name: 'email',
                ...
            },
            {
                displayName: 'Additional Fields',
                // Sets up optional fields
            },
        ],
    };
}

async execute(this: IExecuteFunctions): Promise<INodeExecutionData[][]> {
    let responseData;
    const resource = this.getNodeParameter('resource', 0) as string;
    const operation = this.getNodeParameter('operation', 0) as string;
    //Get credentials the user provided for this node
    const credentials = await this.getCredentials('friendGridApi')
    as IDataObject;
}

```

```

    if (resource === 'contact') {
      if (operation === 'create') {
        // Get email input
        const email = this.getNodeParameter('email', 0) as string;
        // Get additional fields input
        const additionalFields =
          this.getNodeParameter('additionalFields', 0) as IDataObject;
        const data: IDataObject = {
          email,
        };

        Object.assign(data, additionalFields);

        // Make HTTP request as defined in
        // https://sendgrid.com/docs/api-reference/
        const options: IRequestOptions = {
          headers: {
            'Accept': 'application/json',
            'Authorization': `Bearer ${credentials.apiKey}`,
          },
          method: 'PUT',
          body: {
            contacts: [
              data,
            ],
          },
          url: `https://api.sendgrid.com/v3/marketing/contacts`,
          json: true,
        };
        responseData = await this.helpers.httpRequest(options);
      }
    }
    // Map data to n8n data
    return [this.helpers.returnJsonArray(responseData)];
  }
}

```

In declarative style:

```

import { INodeType, INodeTypeDescription } from 'n8n-workflow';

// Create the FriendGrid class
export class FriendGrid implements INodeType {
  description: INodeTypeDescription = {
    displayName: 'FriendGrid',
    name: 'friendGrid',
    ...
    // Set up the basic request configuration
    requestDefaults: {
      baseURL: 'https://api.sendgrid.com/v3/marketing'
    },
    properties: [
      {
        displayName: 'Resource',
        ...
      },
      {
        displayName: 'Operation',
        name: 'operation',
        type: 'options',
      }
    ]
  }
}

```

```

displayOptions: {
  show: [
    resource: [
      'contact',
    ],
  ],
  options: [
    {
      name: 'Create',
      value: 'create',
      description: 'Create a contact',
      // Add the routing object
      routing: {
        request: {
          method: 'POST',
          url: '/contacts',
          send: {
            type: 'body',
            properties: {
              email: {$parameter["email"]}
            }
          }
        }
      }
    },
    // Handle the response to contact creation
    output: {
      postReceive: [
        {
          type: 'set',
          properties: {
            value: '= ${ "success": $response }'
          }
        }
      ]
    }
  ],
  default: 'create',
  description: 'The operation to perform.',
},
{
  displayName: 'Email',
  ...
},
{
  displayName: 'Additional Fields',
  // Sets up optional fields
},
],
}
// No execute method needed
}

```

Design your node's user interface

Most nodes are a GUI (graphical user interface) representation of an API. Designing the interface means finding a user-friendly way to represent API endpoints and parameters. Directly translating an entire API into form fields in a node may not result in a good user experience.

This document provides design guidance and standards to follow. These guidelines are the same as those used by n8n. This helps provide a smooth and consistent user experience for users mixing community and built-in nodes.

Design guidance

All node's use n8n's [node UI elements](#), so you don't need to consider style details such as colors, borders, and so on. However, it's still useful to go through a basic design process:

- Review the documentation for the API you're integrating. Ask yourself:
 - What can you leave out?
 - What can you simplify?
 - Which parts of the API are confusing? How can you help users understand them?
- Use a wireframe tool to try out your field layout. If you find your node has a lot of fields and is getting confusing, consider n8n's guidance on [showing and hiding fields](#).

Standards

UI text style

| Element | Style |
|-----------------|--|
| Drop-down value | Title case |
| Hint | Sentence case |
| Info box | Sentence case. Don't use a period (.) for one-sentence information. Always use a period if there's more than one sentence. This field can include links, which should open in a new tab. |
| Node name | Title case |
| Parameter name | Title case |
| Subtitle | Title case |
| Tooltip | Sentence case. Don't use a period (.) for one-sentence tooltips. Always use a period if there's more than one sentence. This field can include links, which should open in a new tab. |

UI text terminology

- Use the same terminology as the service the node connects to. For example, a Notion node should refer to Notion blocks, not Notion paragraphs, because Notion calls these elements blocks. There are exceptions to this rule, usually to avoid technical terms (for example, refer to the guidance on [name and description for upsert operations](#)).
- Sometimes a service has different terms for something in its API and in its GUI. Use the GUI language in your node, as this is what most users are familiar with. If you think some users may need to refer to the service's API docs, consider including this information in a hint.
- Don't use technical jargon when there are simpler alternatives.
- Be consistent when naming things. For example, choose one of directory or folder then stick to it.

Node naming conventions

| Convention | Correct | Incorrect |
|--|-----------------|------------------------------------|
| If a node is a trigger node, the displayed name should have 'Trigger' at the end, with a space before. | Shopify Trigger | ShopifyTrigger,
Shopify trigger |
| Don't include 'node' in the name. | Asana | Asana Node, Asana node |

Showing and hiding fields

Fields can either be:

- Displayed when the node opens: use this for resources and operations, and required fields.
- Hidden in the **Optional fields** section until a user clicks on that section: use this for optional fields.

Progressively disclose complexity: hide a field until any earlier fields it depends on have values. For example, if you have a **Filter by date** toggle, and a **Date to filter by** datepicker, don't display **Date to filter by** until the user enables **Filter by date**.

Conventions by field type

Credentials

n8n automatically displays credential fields as the top fields in the node.

Resources and operations

APIs usually involve doing something to data. For example, "get all tasks." In this example, "task" is the resource, and "get all" is the operation.

When your node has this resource and operation pattern, your first field should be **Resource**, and your second field should be **Operation**.

Required fields

Order fields by:

- Most important to least important.
- Scope: from broad to narrow. For example, you have fields for **Document**, **Page**, and **Text to insert**, put them in that order.

Optional fields

- Order fields alphabetically. To group similar things together, you can rename them. For example, rename **Email** and **Secondary Email to Email (primary)** and **Email (secondary)**.
- If an optional field has a default value that the node uses when the value isn't set, load the field with that value. Explain this in the field description. For example, **Defaults to false**.
- Connected fields: if one optional fields is dependent on another, bundle them together. They should both be under a single option that shows both fields when selected.
- If you have a lot of optional fields, consider grouping them by theme.

Help

There are five types of help built in to the GUI:

- Info boxes: yellow boxes that appear between fields. Refer to [UI elements | Notice](#) for more information.
 - Use info boxes for essential information. Don't over-use them. By making them rare, they stand out more and grab the user's attention.
- Parameter hints: lines of text displayed beneath a user input field. Use this when there's something the user needs to know, but an info box would be excessive.
- Node hints: provide help in the input panel, output panel, or node details view. Refer to [UI elements | Hints](#) for more information.
- Tooltips: callouts that appear when the user hovers over the tooltip icon [Image: "Screenshot of the tooltip icon. The icon is a ? in a grey circle"]. Use tooltips for extra information that the user might need.
 - You don't have to provide a tooltip for every field. Only add one if it contains useful information.
 - When writing tooltips, think about what the user needs. Don't just copy-paste API parameter descriptions. If the description doesn't make sense, or has errors, improve it.
- Placeholder text: n8n can display placeholder text in a field where the user hasn't entered a value. This can help the user know what's expected in that field.

Info boxes, hints, and tooltips can contain links to more information.

Errors

Make it clear which fields are required.

Add validation rules to fields if possible. For example, check for valid email patterns if the field expects an email.

When displaying errors, make sure only the main error message displays in the red error title. More information should go in **Details**.

Refer to [Node Error Handling](#) for more information.

Toggles

- Tooltips for binary states should start with something like **Whether to . . .**
- You may need a list rather than a toggle:
 - Use toggles when it's clear what happens in a false state. For example, **Simplify Output?**. The alternative (don't simplify output) is clear.
 - Use a dropdown list with named options when you need more clarity. For example, **Append?**. What happens if you don't append is unclear (it could be that nothing happens, or information is overwritten, or discarded).

Lists

- Set default values for lists whenever possible. The default should be the most-used option.
- Sort list options alphabetically.
- You can include list option descriptions. Only add descriptions if they provide useful information.
- If there is an option like **All**, use the word **All**, not shorthand like *********.

Trigger node inputs

When a trigger node has a parameter for specifying which events to trigger on:

- Name the parameter **Trigger on**.
- Don't include a tooltip.

Subtitles

Set subtitles based on the values of the main parameters. For example:

```
subtitle: '={{${parameter["operation"]} + ":" +  
$parameter["resource"]}}',
```

IDs

When performing an operation on a specific record, such as “update a task comment” you need a way to specify which record you want to change.

- Wherever possible, provide two ways to specify a record:
 - By choosing from a pre-populated list. You can generate this list using the `loadOptions` parameter. Refer to [Base files](#) for more information.
 - By entering an ID.
- Name the field <Record name> name or ID. For example, **Workspace Name or ID**. Add a tooltip saying “Choose a name from the list, or specify an ID using an expression.” Link to n8n’s [Expressions](#) documentation.
- Build your node so that it can handle users providing more information than required. For example:
 - If you need a relative path, handle the user pasting in the

- absolute path.
- If the user needs to get an ID from a URL, handle the user pasting in the entire URL.

Dates and timestamps

n8n uses ISO timestamp strings for dates and times. Make sure that any date or timestamp field you add supports all ISO 8601 formats.

JSON

You should support two ways of specifying the content of a text input that expects JSON:

- Typing JSON directly into the text input: you need to parse the resulting string into a JSON object.
- Using an expression that returns JSON.

Node icons

-8<- “_snippets/integrations/node-icons.md”

Common patterns and exceptions

This section provides guidance on handling common design patterns, including some edge cases and exceptions to the main standards.

Simplify responses

APIs can return a lot of data that isn't useful. Consider adding a toggle that allows users to choose to simplify the response data:

- Name: **Simplify Response**
- Description: **Whether to return a simplified version of the response instead of the raw data**

Upsert operations

This should always be a separate operation with:

- Name: **Create or Update**
- Description: **Create a new record, or update the current one if it already exists (upsert)**

Boolean operators

n8n doesn't have good support for combining boolean operators, such as AND and OR, in the GUI. Whenever possible, provide options for all ANDs or all ORs.

For example, you have a field called **Must match** to test if values match. Include options to test for **Any** and **All**, as separate options.

Source keys or binary properties

Binary data is file data, such as spreadsheets or images. In n8n, you need a named key to reference the data. Don't use the terms "binary data" or "binary property" for this field. Instead, use a more

descriptive name: **Input data field name / Output data field name**.

Node file structure

Following best practices and standards in your node structure makes your node easier to maintain. It's helpful if other people need to work with the code.

The file and directory structure of your node depends on:

- Your node's complexity.
- Whether you use node versioning.
- How many nodes you include in the npm package.

n8n recommends using the [n8n-node tool](#) to create the expected node file structure. You can customize the generated scaffolding as required to meet more complex needs.

Required files and directories

Your node must include:

- A `package.json` file at the root of the project. Every npm module requires this.
- A `nodes` directory, containing the code for your node:
 - This directory must contain the [base file](#), in the format `<node-name>.node.ts`. For example, `MyNode.node.ts`.
 - n8n recommends including a [codex file](#), containing metadata for your node. The codex filename must match the node base filename. For example, given a node base file named `MyNode.node.ts`, the codex name is `MyNode.node.json`.
 - The `nodes` directory can contain other files and subdirectories, including directories for versions, and node code split across more than one file to create a modular structure.
- A `credentials` directory, containing your credentials code. This code lives in a single [credentials file](#). The filename format is `<node-name>.credentials.ts`. For example, `MyNode.credentials.ts`.

Modular structure

You can choose whether to place all your node's functionality in one file, or split it out into a base file and other modules, which the base file then imports. Unless your node is very simple, it's a best practice to split it out.

A basic pattern is to separate out operations. Refer to the [HttpBin starter node](#) for an example of this.

For more complex nodes, n8n recommends a directory structure. Refer to the [Airtable node](#) or [Microsoft Outlook node](#) as examples.

- `actions`: a directory containing sub-directories that represent resources.
 - Each sub-directory should contain two types of files:
 - An index file with resource description (named either `<resourceName>.resource.ts` or `index.ts`)

- Files for operations <operationName>.operation.ts. These files should have two exports: description of the operation and an execute function.
- methods: an optional directory dynamic parameters' functions.
- transport: a directory containing the communication implementation.

Versioning

If your node has more than one version, and you're using full versioning, this makes the file structure more complex. You need a directory for each version, along with a base file that sets the default version. Refer to [Node versioning](#) for more information on working with versions, including types of versioning.

Decide how many nodes to include in a package

There are two possible setups when building a node:

- One node in one npm package.
- More than one node in a single npm package.

n8n supports both approaches. If you include more than one node, each node should have its own directory in the nodes directory.

A best-practice example for programmatic nodes

n8n's built-in [Airtable node](#) implements a modular structure and versioning, following recommended patterns.

Build a node

This section provides tutorials on building nodes. It covers:

- [Tutorial: Build a declarative-style node](#)
- [Reference material on file structure](#), parameter definitions for [base](#), [codex](#), and [credentials](#) files, [node UI elements](#), and more.

Coming soon:

- More tutorials
 - Revised guidance on standards
-

Set up your development environment

This document lists the essential dependencies for developing a node, as well as guidance on setting up your editor.

Requirements

To build and test a node, you need:

- Node.js and npm. Minimum version Node 18.17.0. You can find instructions on how to install both using nvm (Node Version Manager) for Linux, Mac, and WSL (Windows Subsystem for Linux) [here](#). For Windows users, refer to Microsoft's guide to [Install NodeJS on Windows](#).
- A local instance of n8n. You can install n8n with `npm install n8n -g`, then follow the steps in [Run your node locally](#) to test your node.
- When [building verified community nodes](#), you must use the [n8n-node tool](#) to create and test your node.

You should also have `git` installed. This allows you to clone and use the [n8n-node-starter](#).

Editor setup

n8n recommends using [VS Code](#) as your editor.

Install these extensions:

- [ESLint](#)
- [EditorConfig](#)
- [Prettier](#)

By using VS Code and these extensions, you get access to the n8n node linter's warnings as you code.

Using the n8n-node tool

The n8n-node tool is the official CLI for developing community nodes for n8n. You can use it to scaffold out new nodes, build your projects, and run your node as you develop it.

Using n8n-node, you can create nodes that adhere to the [guidelines for verified community nodes](#).

Get n8n-node

Run n8n-node without installing

You can create an n8n-node project directly without installing by using the [@n8n/create-node initializer](#) with your package manager:

```
npm create @n8n/node@latest
```

This sets up the initial project files locally (an alternative to installing n8n-node locally and explicitly running the [new command](#)). Afterward, you run the rest of the n8n-node commands through your package

manager's script runner inside the project directory (for example, `npm run dev`).

Install n8n-node globally

You can install n8n-node globally with `npm`:

```
npm install --global @n8n/node-cli
```

Verify access to the command by typing:

```
n8n-node --version
```

Command overview

The n8n-node tool provides the following commands:

new

The new command creates the file system structure and metadata for a new node. This command initializes the same structure as outlined in [run n8n-node without installing](#).

When called, it interactively prompts for details about your project to customize your starting code. You'll provide the project name, choose a node type, and select the starting template that best matches your needs. The n8n-node tool will create your project file structure and optionally install your initial project dependencies.

Learn more about how to use the new command in the [creating a new node section](#).

build

The build command compiles your node and copies all the required assets.

Learn more about how to use the build command in the [building your node section](#).

dev

The dev command runs n8n with your node. It monitors your project directory and automatically rebuilds the live preview when it detects changes.

Learn more about how to use the dev command in the [testing your node in n8n section](#).

lint

The lint command checks the code for the node in the current directory. You can optionally use with the `--fix` option to attempt to automatically fix any issues it identifies.

Learn more about how to use the lint command in the [lint your node section](#).

release

The `release` command publishes your community node package to npm. It uses `release-it` to clean, check and cleanly build your package before publishing it to npm.

Learn more about how to use the `release` command in the [release your node section](#).

Creating a new node

To create a new node with `n8n-node`, call `n8n-node new`. You can call this command entirely interactively or provide details on the command line.

The command will prompt for any missing information about your node and then generate a project structure to get you started. By default, it will follow up by installing the initial project dependencies (you can disable this by passing the `--skip-install` flag).

Setting node details interactively

When called without arguments, `n8n-node new` prompts you for details about your new node interactively:

```
n8n-node new
```

This will start an interactive prompt where you can define the details of your project:

- **What is your node called?** The name of your node. This impacts the name of your project directory, package name, and the n8n node itself. The name must use one of the following formats:
 - `n8n-nodes-<YOUR_NODE_NAME>`
 - `@<YOUR_ORG>/n8n-nodes-<YOUR_NODE_NAME>`
- **What kind of node are you building?** The node type you want to build:
 - **HTTP API:** A low-code, declarative node structure that's designed for faster approval for n8n Cloud.
 - **Other:** A programmatic style node with full flexibility.
- **What template do you want to use?** When using the HTTP API, you can choose the template to start from:
 - **GitHub Issues API:** A demo node that includes multiple operations and credentials. This can help you get familiar with the node structure and conventions.
 - **Start from scratch:** A blank template that will guide you through your custom setup with some further prompts.

When choosing HTTP API > Start from scratch, `n8n-node` will ask you the following:

- **What's the base URL of the API?** The root URL for the API you plan to integrate with.
- **What type of authentication does your API use?** The authentication your node should provide:
 - **API Key:** Send a secret key using headers, query parameters, or the request body.
 - **Bearer Token:** Send a token using the Authorization header (`Authorization: Bearer <token>`).

- **OAuth2**: Use an OAuth 2.0 flow to get access tokens on behalf of a user or app.
- **Basic Auth**: Send the base64-encoded username and password through Authorization headers.
- **Custom**: Create your own credential logic. This will create an empty credential class that you can customize according to your needs.
- **None**: No authentication necessary. Don't create a credential class for the node.

Once you've made your selections, n8n-node will create a new project directory for your node in the current directory. By default, it will also install the initial project dependencies (you can disable this by passing the `--skip-install` flag).

Providing node details on the command line

You can provide some of your node details on the command line to avoid prompts.

You can include the name you want to use for your node an argument:

```
n8n-node new n8n-nodes-myproject
```

If you know the template you want to use ahead of time, you can also pass the value using the `--template` flag:

```
n8n-node new --template declarative/custom
```

The template must be one of the following:

- `declarative/github-issues`: A demo node that includes multiple operations and credentials. This can help you get familiar with the node structure and conventions.
- `declarative/custom`: A blank template that will guide you through your custom setup with some further prompts.
- `programmatic/example`: A programmatic style node with full flexibility.

Building your node

You can build your node by running the `build` command in your project's root directory:

```
n8n-node build
```

n8n-node will compile your TypeScript files and bundle your other project assets. You can also call the `build` script from your package manager. For instance, if you're using `npm`, this works the same:

```
npm run build
```

Lint your node

The n8n-node tool automatically creates a `lint` script for your project as well. You can run with your package manager. For example:

```
n8n-node lint
```

You can also run through your package manager's script runner:

```
npm run lint
```

If you include the `--fix` option (also callable with `npm run lint:fix`), n8n-node will attempt to fix the issues that it identifies:

```
n8n-node lint --fix
```

Testing your node in n8n

To test your node in n8n, you run the `dev` command in your project's root directory:

```
n8n-node dev
```

As with the `build` command, you can also run this through your package manager. For example:

```
npm run dev
```

n8n-node will compile your project and then start up a local n8n instance through `npm` with your node loaded.

Visit your `localhost:5678` to sign in to your n8n instance. If you open a workflow, your node appears in the nodes panel:

[Image: node in nodes panel]

From there, you can add it to your workflow and test the node's functionality as you develop.

Release your node

To publish your node, run the `release` command in your project directory. This command uses `release-it` to build and publish your node.

```
n8n-node release
```

To run with `npm`, type:

```
npm run release
```

When you run the `release` command, n8n-node will perform the following actions:

- build the node
- run lint checks against your files
- update the changelog
- create git tags
- create a GitHub release
- publish the package to npm

Build a declarative-style node

This tutorial walks through building a declarative-style node. Before you begin, make sure this is the node style you need. Refer to [Choose your node building approach](#) for more information.

Prerequisites

You need the following installed on your development machine:

-8<- “_snippets/integrations/creating-nodes/prerequisites.md”

You need some understanding of:

- JavaScript/TypeScript
- REST APIs
- git

Build your node

In this section, you’ll clone n8n’s node starter repository, and build a node that integrates the [NASA API](#). You’ll create a node that uses two of NASA’s services: APOD (Astronomy Picture of the Day) and Mars Rover Photos. To keep the code examples short, the node won’t implement every available option for the Mars Rover Photos endpoint.

Step 1: Set up the project

n8n provides a starter repository for node development. Using the starter ensures you have all necessary dependencies. It also provides a linter.

Clone the repository and navigate into the directory:

1. [Generate a new repository](#) from the template repository.
2. Clone your new repository: shell git clone
`https://github.com/<your-organization>/<your-repo-name>.git n8n-nodes-nasa-pics` cd n8n-nodes-nasa-pics

The starter contains example nodes and credentials. Delete the following directories and files:

- nodes/ExampleNode
- nodes/HTTPBin
- credentials/ExampleCredentials.credentials.ts
- credentials/HttpBinApi.credentials.ts

Now create the following directories and files:

```
nodes/NasaPics
nodes/NasaPics/NasaPics.node.json
nodes/NasaPics/NasaPics.node.ts
credentials/NasaPicsApi.credentials.ts
```

These are the key files required for any node. Refer to [Node file structure](#) for more information on required files and recommended organization.

Now install the project dependencies:

```
npm i
```

Step 2: Add an icon

Save the NASA SVG logo from [here](#) as `nasapics.svg` in `nodes/NasaPics/`.

-8<- `_snippets/integrations/creating-nodes/node-icons.md`

Step 3: Create the node

Every node must have a base file. Refer to [Node base file](#) for detailed information about base file parameters.

In this example, the file is `NasaPics.node.ts`. To keep this tutorial short, you'll place all the node functionality in this one file. When building more complex nodes, you should consider splitting out your functionality into modules. Refer to [Node file structure](#) for more information.

Step 3.1: Imports

Start by adding the import statements:

```
import { INodeType, INodeTypeDescription } from 'n8n-workflow';
```

Step 3.2: Create the main class

The node must export an interface that implements `INodeType`. This interface must include a `description` interface, which in turn contains the `properties` array.

```
export class NasaPics implements INodeType {
    description: INodeTypeDescription = {
        // Basic node details will go here
        properties: [
            // Resources and operations will go here
        ]
    };
}
```

Step 3.3: Add node details

All nodes need some basic parameters, such as their display name, icon, and the basic information for making a request using the node. Add the following to the `description`:

```
displayName: 'NASA Pics',
name: 'nasaPics',
icon: 'file:nasapics.svg',
group: ['transform'],
version: 1,
subtitle: `=${{$parameter["operation"]} + ":" + $parameter["resource"]}}`,
description: 'Get data from NASAs API',
defaults: {
    name: 'NASA Pics',
},
inputs: ['main'],
outputs: ['main'],
credentials: [
{
```

```

        name: 'NasaPicsApi',
        required: true,
    },
],
requestDefaults: {
    baseURL: 'https://api.nasa.gov',
    headers: {
        Accept: 'application/json',
        'Content-Type': 'application/json',
    },
},

```

n8n uses some of the properties set in `description` to render the node in the Editor UI. These properties are `displayName`, `icon`, `description`, and `subtitle`.

Step 3.4: Add resources

The resource object defines the API resource that the node uses. In this tutorial, you're creating a node to access two of NASA's API endpoints: `planetary/apod` and `mars-photos`. This means you need to define two resource options in `NasaPics.node.ts`. Update the `properties` array with the resource object:

```

properties: [
{
    displayName: 'Resource',
    name: 'resource',
    type: 'options',
    noDataExpression: true,
    options: [
        {
            name: 'Astronomy Picture of the Day',
            value: 'astronomyPictureOfDay',
        },
        {
            name: 'Mars Rover Photos',
            value: 'marsRoverPhotos',
        },
    ],
    default: 'astronomyPictureOfDay',
},
// Operations will go here
]

```

`type` controls which UI element n8n displays for the resource, and tells n8n what type of data to expect from the user. `options` results in n8n adding a dropdown that allows users to choose one option. Refer to [Node UI elements](#) for more information.

Step 3.5: Add operations

The operations object defines the available operations on a resource.

In a declarative-style node, the operations object includes `routing` (within the `options` array). This sets up the details of the API call.

Add the following to the `properties` array, after the `resource` object:

```
{

```

```

        displayName: 'Operation',
        name: 'operation',
        type: 'options',
        noDataExpression: true,
        displayOptions: {
            show: [
                resource: [
                    'astronomyPictureOfTheDay',
                ],
            ],
        },
        options: [
            {
                name: 'Get',
                value: 'get',
                action: 'Get the APOD',
                description: 'Get the Astronomy Picture of the day',
                routing: {
                    request: {
                        method: 'GET',
                        url: '/planetary/apod',
                    },
                },
            },
        ],
        default: 'get',
    },
    {
        displayName: 'Operation',
        name: 'operation',
        type: 'options',
        noDataExpression: true,
        displayOptions: {
            show: [
                resource: [
                    'marsRoverPhotos',
                ],
            ],
        },
        options: [
            {
                name: 'Get',
                value: 'get',
                action: 'Get Mars Rover photos',
                description: 'Get photos from the Mars Rover',
                routing: {
                    request: {
                        method: 'GET',
                    },
                },
            },
        ],
        default: 'get',
    },
    {
        displayName: 'Rover name',
        description: 'Choose which Mars Rover to get a photo from',
        required: true,
        name: 'roverName',
        type: 'options',
    }
}

```

```

options: [
    {name: 'Curiosity', value: 'curiosity'},
    {name: 'Opportunity', value: 'opportunity'},
    {name: 'Perseverance', value: 'perseverance'},
    {name: 'Spirit', value: 'spirit'},
],
routing: {
    request: {
        url: '/mars-photos/api/v1/rovers/{{value}}/photos',
    },
},
default: 'curiosity',
displayOptions: {
    show: {
        resource: [
            'marsRoverPhotos',
        ],
    },
},
{
    displayName: 'Date',
    description: 'Earth date',
    required: true,
    name: 'marsRoverDate',
    type: 'dateTime',
    default: '',
    displayOptions: {
        show: {
            resource: [
                'marsRoverPhotos',
            ],
        },
    },
    routing: {
        request: {
            // You've already set up the URL. qs appends the value
            // of the field as a query string
            qs: {
                earth_date: `= ${new
Date($value).toISOString().substr(0,10)}`,
            },
        },
    },
},
// Optional/additional fields will go here

```

This code creates two operations: one to get today's APOD image, and another to send a get request for photos from one of the Mars Rovers. The object named `roverName` requires the user to choose which Rover they want photos from. The `routing` object in the Mars Rover operation references this to create the URL for the API call.

Step 3.6: Optional fields

Most APIs, including the NASA API that you're using in this example, have optional fields you can use to refine your query.

To avoid overwhelming users, n8n displays these under **Additional Fields** in the UI.

For this tutorial, you'll add one additional field, to allow users to pick a date to use with the APOD endpoint. Add the following to the properties array:

```
{
  displayName: 'Additional Fields',
  name: 'additionalFields',
  type: 'collection',
  default: {},
  placeholder: 'Add Field',
  displayOptions: {
    show: [
      resource: [
        'astronomyPictureOfTheDay',
      ],
      operation: [
        'get',
      ],
    ],
  },
  options: [
    {
      displayName: 'Date',
      name: 'apodDate',
      type: 'dateTime',
      default: '',
      routing: {
        request: {
          // You've already set up the URL. qs appends the
          value of the field as a query string
          qs: {
            date: `= ${new
Date($value).toISOString().substr(0,10)}`,
          },
        },
      },
    },
  ],
}
```

Step 4: Set up authentication

The NASA API requires users to authenticate with an API key.

Add the following to `nasaPicsApi.credentials.ts`:

```
import {
  IAuthenticateGeneric,
  ICredentialType,
  INodeProperties,
} from 'n8n-workflow';

export class NasaPicsApi implements ICredentialType {
  name = 'NasaPicsApi';
  displayName = 'NASA Pics API';
  // Uses the link to this tutorial as an example
  // Replace with your own docs links when building your own nodes
  documentationUrl = 'https://docs.n8n.io/integrations/creating-
nodes/build/declarative-style-node/';
  properties: INodeProperties[] = [
```

```

        {
            displayName: 'API Key',
            name: 'apiKey',
            type: 'string',
            default: '',
        },
    ];
    authenticate = {
        type: 'generic',
        properties: {
            qs: {
                'api_key': `=${{$credentials.apiKey}}`
            }
        },
    } as IAuthenticateGeneric;
}

```

For more information about credentials files and options, refer to [Credentials file](#).

Step 5: Add node metadata

Metadata about your node goes in the JSON file at the root of your node. n8n refers to this as the codex file. In this example, the file is `NasaPics.node.json`.

Add the following code to the JSON file:

```

{
    "node": "n8n-nodes-base.NasaPics",
    "nodeVersion": "1.0",
    "codexVersion": "1.0",
    "categories": [
        "Miscellaneous"
    ],
    "resources": {
        "credentialDocumentation": [
            {
                "url": ""
            }
        ],
        "primaryDocumentation": [
            {
                "url": ""
            }
        ]
    }
}

```

For more information on these parameters, refer to [Node codex files](#).

Step 6: Update the npm package details

Your npm package details are in the `package.json` at the root of the project. It's essential to include the `n8n` object with links to the credentials and base node file. Update this file to include the following information:

```
{
    // All node names must start with "n8n-nodes-"
```

```

    "name": "n8n-nodes-nasapics",
    "version": "0.1.0",
    "description": "n8n node to call NASA's APOD and Mars Rover Photo services.",
    "keywords": [
        // This keyword is required for community nodes
        "n8n-community-node-package"
    ],
    "license": "MIT",
    "homepage": "https://n8n.io",
    "author": {
        "name": "Test",
        "email": "test@example.com"
    },
    "repository": {
        "type": "git",
        // Change the git remote to your own repository
        // Add the new URL here
        "url": "git+<your-repo-url>"
    },
    "main": "index.js",
    "scripts": {
        // don't change
    },
    "files": [
        "dist"
    ],
    // Link the credentials and node
    "n8n": {
        "n8nNodesApiVersion": 1,
        "credentials": [
            "dist/credentials/NasaPicsApi.credentials.js"
        ],
        "nodes": [
            "dist/nodes/NasaPics/NasaPics.node.js"
        ]
    },
    "devDependencies": {
        // don't change
    },
    "peerDependencies": {
        // don't change
    }
}

```

You need to update the `package.json` to include your own information, such as your name and repository URL. For more information on npm `package.json` files, refer to [npm's package.json documentation](#).

Test your node

-8<- “_snippets/integrations/creating-nodes/testing.md”

Next steps

- [Deploy your node](#).
- View an example of a declarative node: n8n's [Brevo node](#). Note that the main node is declarative, while the trigger node is in

- programmatic style.
- Learn about [node versioning](#).
-

Build a programmatic-style node

This tutorial walks through building a programmatic-style node. Before you begin, make sure this is the node style you need. Refer to [Choose your node building approach](#) for more information.

Prerequisites

You need the following installed on your development machine:

```
-8<- "_snippets/integrations/creating-nodes/prerequisites.md"
```

You need some understanding of:

- JavaScript/TypeScript
- REST APIs
- git
- [Expressions](#) in n8n

Build your node

In this section, you'll clone n8n's node starter repository, and build a node that integrates the [SendGrid](#). You'll create a node that implements one piece of SendGrid functionality: create a contact.

Step 1: Set up the project

n8n provides a starter repository for node development. Using the starter ensures you have all necessary dependencies. It also provides a linter.

Clone the repository and navigate into the directory:

1. [Generate a new repository](#) from the template repository.
2. Clone your new repository:
shell git clone
 <https://github.com/<your-organization>/<your-repo-name>.git> n8n-nodes-friendgrid cd n8n-nodes-friendgrid

The starter contains example nodes and credentials. Delete the following directories and files:

- nodes/ExampleNode
- nodes/HTTPBin
- credentials/ExampleCredentials.credentials.ts
- credentials/HttpBinApi.credentials.ts

Now create the following directories and files:

```
nodes/FriendGrid  
nodes/FriendGrid/FriendGrid.node.json  
nodes/FriendGrid/FriendGrid.node.ts  
credentials/FriendGridApi.credentials.ts
```

These are the key files required for any node. Refer to [Node file structure](#) for more information on required files and recommended organization.

Now install the project dependencies:

```
npm i
```

Step 2: Add an icon

Save the SendGrid SVG logo from [here](#) as friendGrid.svg in nodes/FriendGrid/.

```
-8<- "_snippets/integrations/creating-nodes/node-icons.md"
```

Step 3: Define the node in the base file

Every node must have a base file. Refer to [Node base file](#) for detailed information about base file parameters.

In this example, the file is FriendGrid.node.ts. To keep this tutorial short, you'll place all the node functionality in this one file. When building more complex nodes, you should consider splitting out your functionality into modules. Refer to [Node file structure](#) for more information.

Step 3.1: Imports

Start by adding the import statements:

```
import {
    IExecuteFunctions,
} from 'n8n-core';

import {
    IDataObject,
    INodeExecutionData,
    INodeType,
    INodeTypeDescription,
    NodeConnectionType
} from 'n8n-workflow';

import {
    OptionsWithUri,
} from 'request';
```

Step 3.2: Create the main class

The node must export an interface that implements INodeType. This interface must include a description interface, which in turn contains the properties array.

```
export class FriendGrid implements INodeType {
    description: INodeTypeDescription = {
        // Basic node details will go here
        properties: [
            // Resources and operations will go here
        ],
    };
}
```

```

    // The execute method will go here
    async execute(this: IExecuteFunctions):
Promise<INodeExecutionData[][]> {
    }
}

```

Step 3.3: Add node details

All programmatic nodes need some basic parameters, such as their display name and icon. Add the following to the description:

```

displayName: 'FriendGrid',
name: 'friendGrid',
icon: 'file:friendGrid.svg',
group: ['transform'],
version: 1,
description: 'Consume SendGrid API',
defaults: {
    name: 'FriendGrid',
},
inputs: [NodeConnectionType.Main],
outputs: [NodeConnectionType.Main],
credentials: [
    {
        name: 'friendGridApi',
        required: true,
    },
],

```

n8n uses some of the properties set in description to render the node in the Editor UI. These properties are displayName, icon, and description.

Step 3.4: Add the resource

The resource object defines the API resource that the node uses. In this tutorial, you're creating a node to access one of SendGrid's API endpoints: /v3/marketing/contacts. This means you need to define a resource for this endpoint. Update the properties array with the resource object:

```

{
    displayName: 'Resource',
    name: 'resource',
    type: 'options',
    options: [
        {
            name: 'Contact',
            value: 'contact',
        },
    ],
    default: 'contact',
    noDataExpression: true,
    required: true,
    description: 'Create a new contact',
},

```

type controls which UI element n8n displays for the resource, and tells n8n what type of data to expect from the user. options results in n8n adding a dropdown that allows users to choose one option. Refer to

[Node UI elements](#) for more information.

Step 3.5: Add operations

The operations object defines what you can do with a resource. It usually relates to REST API verbs (GET, POST, and so on). In this tutorial, there's one operation: create a contact. It has one required field, the email address for the contact the user creates.

Add the following to the properties array, after the resource object:

```
{  
  displayName: 'Operation',  
  name: 'operation',  
  type: 'options',  
  displayOptions: {  
    show: {  
      resource: [  
        'contact'  
      ],  
    },  
  },  
  options: [  
    {  
      name: 'Create',  
      value: 'create',  
      description: 'Create a contact',  
      action: 'Create a contact',  
    },  
  ],  
  default: 'create',  
  noDataExpression: true,  
},  
{  
  displayName: 'Email',  
  name: 'email',  
  type: 'string',  
  required: true,  
  displayOptions: {  
    show: {  
      operation: [  
        'create'  
      ],  
      resource: [  
        'contact'  
      ],  
    },  
  },  
  default: '',  
  placeholder: 'name@email.com',  
  description: 'Primary email for the contact',  
},
```

Step 3.6: Add optional fields

Most APIs, including the SendGrid API that you're using in this example, have optional fields you can use to refine your query.

To avoid overwhelming users, n8n displays these under **Additional Fields** in the UI.

For this tutorial, you'll add two additional fields, to allow users to enter the contact's first name and last name. Add the following to the properties array:

```
{
  displayName: 'Additional Fields',
  name: 'additionalFields',
  type: 'collection',
  placeholder: 'Add Field',
  default: {},
  displayOptions: {
    show: [
      resource: [
        'contact',
      ],
      operation: [
        'create',
      ],
    ],
  },
  options: [
    {
      displayName: 'First Name',
      name: 'firstName',
      type: 'string',
      default: '',
    },
    {
      displayName: 'Last Name',
      name: 'lastName',
      type: 'string',
      default: '',
    },
  ],
},
```

Step 4: Add the execute method

You've set up the node UI and basic information. It's time to map the node UI to API requests, and make the node actually do something.

The execute method runs every time the node runs. In this method, you have access to the input items and to the parameters that the user set in the UI, including the credentials.

Add the following the execute method in the FriendGrid.node.ts:

```
// Handle data coming from previous nodes
const items = this.getInputData();
let responseData;
const returnData = [];
const resource = this.getNodeParameter('resource', 0) as string;
const operation = this.getNodeParameter('operation', 0) as string;

// For each item, make an API call to create a contact
for (let i = 0; i < items.length; i++) {
  if (resource === 'contact') {
    if (operation === 'create') {
      // Get email input
      const email = this.getNodeParameter('email', i) as
```

```

string;
    // Get additional fields input
    const additionalFields =
this.getNodeParameter('additionalFields', i) as IDataObject;
    const data: IDataObject = {
        email,
    };

    Object.assign(data, additionalFields);

    // Make HTTP request according to
    https://sendgrid.com/docs/api-reference/
    const options: OptionsWithUri = {
        headers: {
            'Accept': 'application/json',
        },
        method: 'PUT',
        body: {
            contacts: [
                data,
            ],
        },
        uri:
`https://api.sendgrid.com/v3/marketing/contacts`,
        json: true,
    };
    responseData = await
this.helpers.requestWithAuthentication.call(this, 'friendGridApi',
options);
    returnData.push(responseData);
}
}
}
// Map data to n8n data structure
return [this.helpers.returnJsonArray(returnData)];

```

Note the following lines of this code:

```

const items = this.getInputData();
...
for (let i = 0; i < items.length; i++) {
    ...
    const email = this.getNodeParameter('email', i) as string;
    ...
}

```

Users can provide data in two ways:

- Entered directly in the node fields
- By mapping data from earlier nodes in the workflow

`getInputData()`, and the subsequent loop, allows the node to handle situations where data comes from a previous node. This includes supporting multiple inputs. This means that if, for example, the previous node outputs contact information for five people, your FriendGrid node can create five contacts.

Step 5: Set up authentication

The SendGrid API requires users to authenticate with an API key.

Add the following to `FriendGridApi.credentials.ts`

```
import {
    IAuthenticateGeneric,
    ICredentialTestRequest,
    ICredentialType,
    INodeProperties,
} from 'n8n-workflow';

export class FriendGridApi implements ICredentialType {
    name = 'friendGridApi';
    displayName = 'FriendGrid API';
    properties: INodeProperties[] = [
        {
            displayName: 'API Key',
            name: 'apiKey',
            type: 'string',
            default: '',
        },
    ];
    authenticate: IAuthenticateGeneric = {
        type: 'generic',
        properties: {
            headers: {
                Authorization: '=Bearer {{$credentials.apiKey}}',
            },
        },
    };
    test: ICredentialTestRequest = {
        request: {
            baseURL: 'https://api.sendgrid.com/v3',
            url: '/marketing/contacts',
        },
    };
}
```

For more information about credentials files and options, refer to [Credentials file](#).

Step 6: Add node metadata

Metadata about your node goes in the JSON file at the root of your node. n8n refers to this as the codex file. In this example, the file is `FriendGrid.node.json`.

Add the following code to the JSON file:

```
{
    "node": "n8n-nodes-base.FriendGrid",
    "nodeVersion": "1.0",
    "codexVersion": "1.0",
    "categories": [
        "Miscellaneous"
    ],
    "resources": {
        "credentialDocumentation": [
            {
                "url": ""
            }
        ]
    }
}
```

```

        }
    ],
    "primaryDocumentation": [
        {
            "url": ""
        }
    ]
}

```

For more information on these parameters, refer to [Node codex files](#).

Step 7: Update the npm package details

Your npm package details are in the `package.json` at the root of the project. It's essential to include the `n8n` object with links to the credentials and base node file. Update this file to include the following information:

```
{
    // All node names must start with "n8n-nodes-"
    "name": "n8n-nodes-friendgrid",
    "version": "0.1.0",
    "description": "n8n node to create contacts in SendGrid",
    "keywords": [
        // This keyword is required for community nodes
        "n8n-community-node-package"
    ],
    "license": "MIT",
    "homepage": "https://n8n.io",
    "author": {
        "name": "Test",
        "email": "test@example.com"
    },
    "repository": {
        "type": "git",
        // Change the git remote to your own repository
        // Add the new URL here
        "url": "git+<your-repo-url>"
    },
    "main": "index.js",
    "scripts": {
        // don't change
    },
    "files": [
        "dist"
    ],
    // Link the credentials and node
    "n8n": {
        "n8nNodesApiVersion": 1,
        "credentials": [
            "dist/credentials/FriendGridApi.credentials.js"
        ],
        "nodes": [
            "dist/nodes/FriendGrid/FriendGrid.node.js"
        ]
    },
    "devDependencies": {
        // don't change
    },
}
```

```
"peerDependencies": {  
    // don't change  
}  
}
```

You need to update the `package.json` to include your own information, such as your name and repository URL. For more information on npm `package.json` files, refer to [npm's package.json documentation](#).

Test your node

-8<- `_snippets/integrations/creating-nodes/testing.md`

Next steps

- [Deploy your node](#).
 - View an example of a programmatic node: n8n's [Mattermost node](#). This is an example of a more complex programmatic node structure.
 - Learn about [node versioning](#).
 - Make sure you understand key concepts: [item linking](#) and [data structures](#).
-

Node building reference

This section contains reference information, including details about:

- [Node UI elements](#)
 - [Organizing your node files](#)
 - Key parameters in your node's [base file](#) and [credentials file](#).
 - [UX guidelines](#) and [verification guidelines](#) for submitting your node for [verification by n8n](#).
-

Node user interface elements

n8n provides a set of predefined UI components (based on a JSON file) that allows users to input all sorts of data types. The following UI elements are available in n8n.

String

Basic configuration:

```
{  
  displayName: Name, // The value the user sees in the UI  
  name: name, // The name used to reference the element UI within  
  the code  
  type: string,  
  required: true, // Whether the field is required or not  
  default: 'n8n',  
  description: 'The name of the user',
```

```

    displayOptions: { // the resources and operations to display
this element with
        show: {
            resource: [
                // comma-separated list of resource names
            ],
            operation: [
                // comma-separated list of operation names
            ]
        }
    },
}

```

[Image: String]

String field for inputting passwords:

```

{
    displayName: 'Password',
    name: 'password',
    type: 'string',
    required: true,
    typeOptions: {
        password: true,
    },
    default: '',
    description: `User's password`,
    displayOptions: { // the resources and operations to display
this element with
        show: {
            resource: [
                // comma-separated list of resource names
            ],
            operation: [
                // comma-separated list of operation names
            ]
        }
    },
}

```

[Image: Password]

String field with more than one row:

```

{
    displayName: 'Description',
    name: 'description',
    type: 'string',
    required: true,
    typeOptions: {
        rows: 4,
    },
    default: '',
    description: 'Description',
    displayOptions: { // the resources and operations to display
this element with
        show: {
            resource: [
                // comma-separated list of resource names
            ],
            operation: [

```

```

        // comma-separated list of operation names
    ]
}
},
}
}

```

[Image: Multiple rows]

Support drag and drop for data keys

Users can drag and drop data values to map them to fields. Dragging and dropping creates an expression to load the data value. n8n supports this automatically.

You need to add an extra configuration option to support dragging and dropping data keys:

- `requiresDataPath: 'single'`: for fields that require a single string.
- `requiresDataPath: 'multiple'`: for fields that can accept a comma-separated list of string.

The [Compare Datasets node](#) code has examples.

Number

Number field with decimal points:

```

{
  displayName: 'Amount',
  name: 'amount',
  type: 'number',
  required: true,
  typeOptions: {
    maxValue: 10,
    minValue: 0,
    numberPrecision: 2,
  },
  default: 10.00,
  description: 'Your current amount',
  displayOptions: { // the resources and operations to display
this element with
    show: {
      resource: [
        // comma-separated list of resource names
      ],
      operation: [
        // comma-separated list of operation names
      ]
    }
  },
}

```

[Image: Decimal]

Collection

Use the collection type when you need to display optional fields.

```

{
  displayName: 'Filters',
  name: 'filters',
  type: 'collection',
  placeholder: 'Add Field',
  default: {},
  options: [
    {
      displayName: 'Type',
      name: 'type',
      type: 'options',
      options: [
        {
          name: 'Automated',
          value: 'automated',
        },
        {
          name: 'Past',
          value: 'past',
        },
        {
          name: 'Upcoming',
          value: 'upcoming',
        },
      ],
      default: '',
    },
  ],
  displayOptions: { // the resources and operations to display
this element with
  show: {
    resource: [
      // comma-separated list of resource names
    ],
    operation: [
      // comma-separated list of operation names
    ]
  }
},
}

```

[Image: Collection]

DateTime

The `dateTime` type provides a date picker.

```

{
  displayName: 'Modified Since',
  name: 'modified_since',
  type: 'dateTime',
  default: '',
  description: 'The date and time when the file was last
modified',
  displayOptions: { // the resources and operations to display
this element with
  show: {
    resource: [
      // comma-separated list of resource names
    ]
  }
},
}

```

```

        ],
        operation: [
            // comma-separated list of operation names
        ]
    },
},
}

```

[Image: DateTime]

Boolean

The boolean type adds a toggle for entering true or false.

```

{
    displayName: 'Wait for Image',
    name: 'waitForImage',
    type: 'boolean',
    default: true, // Initial state of the toggle
    description: 'Whether to wait for the image or not',
    displayOptions: { // the resources and operations to display
this element with
        show: {
            resource: [
                // comma-separated list of resource names
            ],
            operation: [
                // comma-separated list of operation names
            ]
        }
    },
}

```

[Image: Boolean]

Color

The color type provides a color selector.

```

{
    displayName: 'Background Color',
    name: 'backgroundColor',
    type: 'color',
    default: '', // Initially selected color
    displayOptions: { // the resources and operations to display
this element with
        show: {
            resource: [
                // comma-separated list of resource names
            ],
            operation: [
                // comma-separated list of operation names
            ]
        }
    },
}

```

[Image: Color]

Options

The `options` type adds an options list. Users can select a single value.

```
{  
  displayName: 'Resource',  
  name: 'resource',  
  type: 'options',  
  options: [  
    {  
      name: 'Image',  
      value: 'image',  
    },  
    {  
      name: 'Template',  
      value: 'template',  
    },  
  ],  
  default: 'image', // The initially selected option  
  description: 'Resource to consume',  
  displayOptions: { // the resources and operations to display  
    this element with  
    show: {  
      resource: [  
        // comma-separated list of resource names  
      ],  
      operation: [  
        // comma-separated list of operation names  
      ]  
    }  
  },  
}
```

[Image: Options]

Multi-options

The `multiOptions` type adds an options list. Users can select more than one value.

```
{  
  displayName: 'Events',  
  name: 'events',  
  type: 'multiOptions',  
  options: [  
    {  
      name: 'Plan Created',  
      value: 'planCreated',  
    },  
    {  
      name: 'Plan Deleted',  
      value: 'planDeleted',  
    },  
  ],  
  default: [], // Initially selected options  
  description: 'The events to be monitored',  
  displayOptions: { // the resources and operations to display  
    this element with  
    show: {
```

```

        resource: [
            // comma-separated list of resource names
        ],
        operation: [
            // comma-separated list of operation names
        ]
    }
},
}

```

[Image: Multi-options]

Filter

Use this component to evaluate, match, or filter incoming data.

This is the code from n8n's own If node. It shows a filter component working with a [collection](#) component where users can configure the filter's behavior.

```

{
    displayName: 'Conditions',
    name: 'conditions',
    placeholder: 'Add Condition',
    type: 'filter',
    default: {},
    typeOptions: {
        filter: {
            // Use the user options (below) to determine filter
            behavior
            caseSensitive: '={!$parameter.options.ignoreCase}',
            typeValidation: '=
{$parameter.options.looseTypeValidation ? "loose" : "strict"}',
            },
        },
        {
        displayName: 'Options',
        name: 'options',
        type: 'collection',
        placeholder: 'Add option',
        default: {},
        options: [
            {
                displayName: 'Ignore Case',
                description: 'Whether to ignore letter case when evaluating
conditions',
                name: 'ignoreCase',
                type: 'boolean',
                default: true,
            },
            {
                displayName: 'Less Strict Type Validation',
                description: 'Whether to try casting value types based on
the selected operator',
                name: 'looseTypeValidation',
                type: 'boolean',
                default: true,
            },
        ],
    }
}

```

```
],  
},
```

[Image: Filter]

Assignment collection (drag and drop)

Use the drag and drop component when you want users to pre-fill name and value parameters with a single drag interaction.

```
{  
  displayName: 'Fields to Set',  
  name: 'assignments',  
  type: 'assignmentCollection',  
  default: {},  
},
```

You can see an example in n8n's [Edit Fields \(Set\) node](#):

[Image: A gif showing the drag and drop action, as well as changing a field to fixed]

Fixed collection

Use the `fixedCollection` type to group fields that are semantically related.

```
{  
  displayName: 'Metadata',  
  name: 'metadataUi',  
  placeholder: 'Add Metadata',  
  type: 'fixedCollection',  
  default: '',  
  typeOptions: {  
    multipleValues: true,  
  },  
  description: '',  
  options: [  
    {  
      name: 'metadataValues',  
      displayName: 'Metadata',  
      values: [  
        {  
          displayName: 'Name',  
          name: 'name',  
          type: 'string',  
          default: 'Name of the metadata key to add.',  
        },  
        {  
          displayName: 'Value',  
          name: 'value',  
          type: 'string',  
          default: '',  
          description: 'Value to set for the metadata  
key.',  
        },  
      ],  
    },  
  ],  
},
```

```

        ],
        displayOptions: { // the resources and operations to display
this element with
            show: {
                resource: [
                    // comma-separated list of resource names
                ],
                operation: [
                    // comma-separated list of operation names
                ]
            }
        },
    }
}

```

[Image: Fixed collection]

Resource locator

[Image: Resource locator]

The resource locator element helps users find a specific resource in an external service, such as a card or label in Trello.

The following options are available:

- ID
- URL
- List: allows users to select or search from a prepopulated list. This option requires more coding, as you must populate the list, and handle searching if you choose to support it.

You can choose which types to include.

Example:

```
{
    displayName: 'Card',
    name: 'cardID',
    type: 'resourceLocator',
    default: '',
    description: 'Get a card',
    modes: [
        {
            displayName: 'ID',
            name: 'id',
            type: 'string',
            hint: 'Enter an ID',
            validation: [
                {
                    type: 'regex',
                    properties: {
                        regex: '^[0-9]',
                        errorMessage: 'The ID must start with a
number',
                    },
                },
            ],
            placeholder: '12example',
            // How to use the ID in API call
            url: '=http://api-base-url.com/?id={{$value}}',
        }
    ]
}
```

```

},
{
  displayName: 'URL',
  name: 'url',
  type: 'string',
  hint: 'Enter a URL',
  validation: [
    {
      type: 'regex',
      properties: {
        regex: '^http',
        errorMessage: 'Invalid URL',
      },
    },
  ],
  placeholder: 'https://example.com/card/12example/',
  // How to get the ID from the URL
  extractValue: {
    type: 'regex',
    regex: 'example.com/card/([0-9]*.*)/',
  },
},
{
  displayName: 'List',
  name: 'list',
  type: 'list',
  typeOptions: {
    // You must always provide a search method
    // Write this method within the methods object in
    // your base file
    // The method must populate the list, and handle
    searchingIfSearchable: true
    searchListMethod: 'searchMethod',
    // If you want users to be able to search the list
    searchable: true,
    // Set to true if you want to force users to search
    // When true, users can't browse the list
    // Or false if users can browse a list
    searchFilterRequired: true,
  },
},
],
displayOptions: {
  // the resources and operations to display this element with
  show: {
    resource: [
      // comma-separated list of resource names
    ],
    operation: [
      // comma-separated list of operation names
    ],
  },
},
}

```

Refer to the following for live examples:

- Refer to [CardDescription.ts](#) and [Trello.node.ts](#) in n8n's Trello node for an example of a list with search that includes `searchFilterRequired: true`.
- Refer to [GoogleDrive.node.ts](#) for an example where users can

browse the list or search.

Resource mapper

If your node performs insert, update, or upsert operations, you need to send data from the node in a format supported by the service you're integrating with. A common pattern is to use a Set node before the node that sends data, to convert 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, rather than using a Set node. The resource mapper component can also validate input data against the schema provided in the node, and cast input data into the expected type.

```
{
    displayName: 'Columns',
    name: 'columns', // The name used to reference the element UI
within the code
    type: 'resourceMapper', // The UI element type
    default: {
        // mappingMode can be defined in the component (mappingMode:
'defineBelow')
        // or you can attempt automatic mapping (mappingMode:
'autoMapInputData')
        mappingMode: 'defineBelow',
        // Important: always set default value to null
        value: null,
    },
    required: true,
    // See "Resource mapper type options interface" below for the
full typeOptions specification
    typeOptions: {
        resourceMapper: {
            resourceMapperMethod: 'getMappingColumns',
            mode: 'update',
            fieldWords: {
                singular: 'column',
                plural: 'columns',
            },
            addAllFields: true,
            multiKeyMatch: true,
            supportAutoMap: true,
            matchingFieldsLabels: {
                title: 'Custom matching columns title',
                description: 'Help text for custom matching
columns',
                hint: 'Below-field hint for custom matching
columns',
            },
        },
    },
},
```

Refer to the [Postgres node \(version 2\)](#) for a live example using a database schema.

Refer to the [Google Sheets node \(version 2\)](#) for a live example using a schema-less service.

Resource mapper type options interface

The typeOptions section must implement the following interface:

```
export interface ResourceMapperTypeOptions {
    // The name of the method where you fetch the schema
    // Refer to the Resource mapper method section for more detail
    resourceMapperMethod: string;
    // Choose the mode for your operation
    // Supported modes: add, update, upsert
    mode: 'add' | 'update' | 'upsert';
    // Specify labels for fields in the UI
    fieldWords?: { singular: string; plural: string };
    // Whether n8n should display a UI input for every field when
    node first added to workflow
    // Default is true
    addAllFields?: boolean;
    // Specify a message to show if no fields are fetched from the
    service
    // (the call is successful but the response is empty)
    noFieldsError?: string;
    // Whether to support multi-key column matching
    // multiKeyMatch is for update and upsert only
    // Default is false
    // If true, the node displays a multi-select dropdown for the
    matching column selector
    multiKeyMatch?: boolean;
    // Whether to support automatic mapping
    // If false, n8n hides the mapping mode selector field and sets
    mappingMode to defineBelow
    supportAutoMap?: boolean;
    // Custom labels for the matching columns selector
    matchingFieldsLabels?: {
        title?: string;
        description?: string;
        hint?: string;
    };
}
```

Resource mapper method

This method contains your node-specific logic for fetching the data schema. Every node must implement its own logic for fetching the schema, and setting up each UI field according to the schema.

It must return a value that implements the ResourceMapperFields interface:

```
interface ResourceMapperField {
    // Field ID as in the service
    id: string;
    // Field label
    displayName: string;
    // Whether n8n should pre-select the field as a matching field
    // A matching field is a column used to identify the rows to
    modify
    defaultMatch: boolean;
    // Whether the field can be used as a matching field
    canBeUsedToMatch?: boolean;
    // Whether the field is required by the schema
```

```

    required: boolean;
    // Whether to display the field in the UI
    // If false, can't be used for matching or mapping
    display: boolean;
    // The data type for the field
    // These correspond to UI element types
    // Supported types: string, number, date, boolean, time,
array, object, options
    type?: FieldType;
    // Added at runtime if the field is removed from mapping by the
user
    removed?: boolean;
    // Specify options for enumerated types
    options?: INodePropertyOptions[];
}

```

Refer to the [Postgres resource mapping method](#) and [Google Sheets resource mapping method](#) for live examples.

JSON

```

{
  displayName: 'Content (JSON)',
  name: 'content',
  type: 'json',
  default: '',
  description: '',
  displayOptions: { // the resources and operations to display
this element with
  show: {
    resource: [
      // comma-separated list of resource names
    ],
    operation: [
      // comma-separated list of operation names
    ]
  }
},
}

```

[Image: JSON]

HTML

The HTML editor allows users to create HTML templates in their workflows. The editor supports standard HTML, CSS in `<style>` tags, and expressions wrapped in `{}{}`. Users can add `<script>` tags to pull in additional JavaScript. n8n doesn't run this JavaScript during workflow execution.

```

{
  displayName: 'HTML Template', // The value the user sees in the
UI
  name: 'html', // The name used to reference the element UI
within the code
  type: 'string',
  typeOptions: {
    editor: 'htmlEditor',
  }
}

```

```

},
  default: placeholder, // Loads n8n's placeholder HTML template
  noDataExpression: true, // Prevent using an expression for the
field
  description: 'HTML template to render',
},

```

Refer to [Html.node.ts](#) for a live example.

Notice

Display a yellow box with a hint or extra info. Refer to [Node UI design](#) for guidance on writing good hints and info text.

```

{
  displayName: 'Your text here',
  name: 'notice',
  type: 'notice',
  default: '',
},

```

[Image: Notice]

Hints

There are two types of hints: parameter hints and node hints:

- Parameter hints are small lines of text below a user input field.
- Node hints are a more powerful and flexible option than [Notice](#). Use them to display longer hints, in the input panel, output panel, or node details view.

Add a parameter hint

Add the `hint` parameter to a UI element:

```

{
  displayName: 'URL',
  name: 'url',
  type: 'string',
  hint: 'Enter a URL',
  ...
}

```

Add a node hint

Define the node's hints in the `hints` property within the node `description`:

```

description: INodeTypeDescription = {
  ...
  hints: [
    {
      // The hint message. You can use HTML.
      message: "This node has many input items. Consider
enabling <b>Execute Once</b> in the node's settings.",
      // Choose from: info, warning, danger. The default is
    }
  ]
}

```

```

'info'.
    // Changes the color. info (grey), warning (yellow),
danger (red)
    type: 'info',
        // Choose from: inputPane, outputPane, ndv. By default
n8n displays the hint in both the input and output panels.
    location: 'outputPane',
        // Choose from: always, beforeExecution, afterExecution.
The default is 'always'
    whenToDisplay: 'beforeExecution',
        // Optional. An expression. If it resolves to true, n8n
displays the message. Defaults to true.
    displayCondition: '= ${parameter["operation"] ===
"select" && $input.all().length > 1 }}'
}
]
...
}

```

Add a dynamic hint to a programmatic-style node

In programmatic-style nodes you can create a dynamic message that includes information from the node execution. As it relies on the node output data, you can't display this type of hint until after execution.

```

if (operation === 'select' && items.length > 1 && !node.executeOnce)
{
    // Expects two parameters: NodeExecutionData and an array of
hints
    return new NodeExecutionOutput(
        [returnData],
        [
            {
                message: `This node ran ${items.length} times, once
for each input item. To run for the first item only, enable
<b>Execute once</b> in the node settings.`,
                location: 'outputPane',
            },
        ],
    );
}
return [returnData];

```

For a live example of a dynamic hint in a programmatic-style node, view the [Split Out node code](#).

Code standards

Following defined code standards when building your node makes your code more readable and maintainable, and helps avoid errors. This document provides guidance on good code practices for node building. It focuses on code details. For UI standards and UX guidance, refer to [Node UI design](#).

Use the linter

The n8n node linter provides automatic checking for many of the node-building standards. You should ensure your node passes the linter's checks before publishing it. Refer to the [n8n node linter](#) documentation for more information.

Use the starter

The n8n node starter project includes a recommended setup, dependencies (including the linter), and examples to help you get started. Begin new projects with the [starter](#).

Write in TypeScript

All n8n code is TypeScript. Writing your nodes in TypeScript can speed up development and reduce bugs.

Detailed guidelines for writing a node

These guidelines apply to any node you build.

Resources and operations

If your node can perform several operations, call the parameter that sets the operation `Operation`. If your node can do these operations on more than one resource, create a `Resource` parameter. The following code sample shows a basic resource and operations setup:

```
export const ExampleNode implements INodeType {
    description: {
        displayName: 'Example Node',
        ...
    },
    properties: [
        {
            displayName: 'Resource',
            name: 'resource',
            type: 'options',
            options: [
                {
                    name: 'Resource One',
                    value: 'resourceOne'
                },
                {
                    name: 'Resource Two',
                    value: 'resourceTwo'
                }
            ],
            default: 'resourceOne'
        },
        {
            displayName: 'Operation',
            name: 'operation',
            type: 'options',
            // Only show these operations for Resource One
            displayOptions: {
                show: {
                    resource: [

```

```
        'resourceOne'
    ]
}
},
options: [
{
    name: 'Create',
    value: 'create',
    description: 'Create an instance of Resource
One'
}
]
}
]
```

Reuse internal parameter names

All resource and operation fields in an n8n node have two settings: a display name, set using the `name` parameter, and an internal name, set using the `value` parameter. Reusing the internal name for fields allows n8n to preserve user-entered data if a user switches operations.

For example: you're building a node with a resource named 'Order'. This resource has several operations, including Get, Edit, and Delete. Each of these operations uses an order ID to perform the operation on the specified order. You need to display an ID field for the user. This field has a display label, and an internal name. By using the same internal name (set in value) for the operation ID field on each resource, a user can enter the ID with the Get operation selected, and not lose it if they switch to Edit.

When reusing the internal name, you must ensure that only one field is visible to the user at a time. You can control this using `displayOptions`.

Detailed guidelines for writing a programmatic-style node

These guidelines apply when building nodes using the programmatic node-building style. They aren't relevant when using the declarative style. For more information on different node-building styles, refer to [Choose your node building approach](#).

Don't change incoming data

Never change the incoming data a node receives (data accessible with `this.getInputData()`) as all nodes share it. If you need to add, change, or delete data, clone the incoming data and return the new data. If you don't do this, sibling nodes that execute after the current one will operate on the altered data and process incorrect data.

It's not necessary to always clone all the data. For example, if a node changes the binary data but not the JSON data, you can create a new item that reuses the reference to the JSON item.

Use the built in request library

Some third-party services have their own libraries on npm, which make it easier to create an integration. The problem with these packages is that you add another dependency (plus all the dependencies of the dependencies). This adds more and more code, which has to be loaded, can introduce security vulnerabilities, bugs, and so on. Instead, use the built-in module:

```
// If no auth needed
const response = await this.helpers.httpRequest(options);

// If auth needed
const response = await
this.helpers.httpRequestWithAuthentication.call(
  this,
  'credentialTypeName', // For example: pipedriveApi
  options,
);
```

This uses the npm package [Axios](#).

Refer to [HTTP helpers](#) for more information, and for migration instructions for the removed `this.helpers.request`.

Error handling in n8n nodes

Proper error handling is crucial for creating robust n8n nodes that provide clear feedback to users when things go wrong. n8n provides two specialized error classes to handle different types of failures in node implementations:

- [NodeApiError](#): For API-related errors and external service failures
- [NodeOperationError](#): For operational errors, validation failures, and configuration issues

NodeApiError

Use `NodeApiError` when dealing with external API calls and HTTP requests. This error class is specifically designed to handle API response errors and provides enhanced features for parsing and presenting API-related failures such as:

- HTTP request failures
- external API errors
- authentication/authorization failures
- rate limiting errors
- service unavailable errors

Initialize new `NodeApiError` instances using the following pattern:

```
new NodeApiError(node: INode, errorResponse: JsonObject, options?: NodeApiErrorOptions)
```

Common usage patterns

For basic API request failures, catch the error and wrap it in `NodeApiError`:

```
try {
    const response = await
this.helpers.requestWithAuthentication.call(
    this,
    credentialType,
    options
);
    return response;
} catch (error) {
    throw new NodeApiError(this.getNode(), error as JsonObject);
}
```

Handle specific HTTP status codes with custom messages:

```
try {
    const response = await
this.helpers.requestWithAuthentication.call(
    this,
    credentialType,
    options
);
    return response;
} catch (error) {
    if (error.httpCode === "404") {
        const resource = this.getNodeParameter("resource", 0) as
string;
        const errorOptions = {
            message: `${resource.charAt(0).toUpperCase() + resource.slice(1)
} not found`,
            description:
                "The requested resource could not be found. Please
check your input parameters.",
        };
        throw new NodeApiError(
            this.getNode(),
            error as JsonObject,
            errorOptions
        );
    }

    if (error.httpCode === "401") {
        throw new NodeApiError(this.getNode(), error as JsonObject,
{
            message: "Authentication failed",
            description: "Please check your credentials and try
again.",
        });
    }

    throw new NodeApiError(this.getNode(), error as JsonObject);
}
```

NodeOperationError

Use `NodeOperationError` for:

- operational errors
- validation failures
- configuration issues that aren't related to external API calls
- input validation errors
- missing required parameters
- data transformation errors
- workflow logic errors

Initialize new `NodeOperationError` instances using the following pattern:

```
new NodeOperationError(node: INode, error: Error | string | JsonObject, options?: NodeOperationErrorOptions)
```

Common usage patterns

Use `NodeOperationError` for validating user inputs:

```
const email = this.getNodeParameter("email", itemIndex) as string;

if (email.indexOf("@") === -1) {
  const description = `The email address '${email}' in the 'email' field isn't valid`;
  throw new NodeOperationError(this.getNode(), "Invalid email address", {
    description,
    itemIndex, // for multiple items, this will link the error to the specific item
  });
}
```

When processing multiple items, include the item index for better error context:

```
for (let i = 0; i < items.length; i++) {
  try {
    // Process item
    const result = await processItem(items[i]);
    returnData.push(result);
  } catch (error) {
    if (this.continueOnFail()) {
      returnData.push({
        json: { error: error.message },
        pairedItem: { item: i },
      });
      continue;
    }
  }
}

throw new NodeOperationError(this.getNode(), error as Error,
{
  description: error.description,
  itemIndex: i,
});
```

Node versioning

n8n supports node versioning. You can make changes to existing nodes without breaking the existing behavior by introducing a new version.

Be aware of how n8n decides which node version to load:

- If a user builds and saves a workflow using version 1, n8n continues to use version 1 in that workflow, even if you create and publish a version 2 of the node.
- When a user creates a new workflow and browses for nodes, n8n always loads the latest version of the node.

Light versioning

This is available for all node types.

One node can contain more than one version, allowing small version increments without code duplication. To use this feature:

1. Change the main `version` parameter to an array, and add your version numbers, including your existing version.
2. You can then access the `version` parameter with `@version` in your `displayOptions` in any object (to control which versions n8n displays the object with). You can also query the version from a function using `const nodeVersion = this.getNode().typeVersion;`.

As an example, say you want to add versioning to the NasaPics node from the [Declarative node tutorial](#), then configure a resource so that n8n only displays it in version 2 of the node. In your base `NasaPics.node.ts` file:

```
{
  displayName: 'NASA Pics',
  name: 'NasaPics',
  icon: 'file:nasapics.svg',
  // List the available versions
  version: [1,2,3],
  // More basic parameters here
  properties: [
    // Add a resource that's only displayed for version2
    {
      displayName: 'Resource name',
      // More resource parameters
      displayOptions: {
        show: {
          '@version': 2,
        },
      },
    },
  ],
}
```

Full versioning

This isn't available for declarative-style nodes.

As an example, refer to the [Mattermost node](#).

Full versioning summary:

- The base node file should extend `NodeVersionedType` instead of `INodeType`.
 - The base node file should contain a description including the `defaultVersion` (usually the latest), other basic node metadata such as name, and a list of versions. It shouldn't contain any node functionality.
 - n8n recommends using `v1`, `v2`, and so on, for version folder names.
-

Node file structure

Following best practices and standards in your node structure makes your node easier to maintain. It's helpful if other people need to work with the code.

The file and directory structure of your node depends on:

- Your node's complexity.
- Whether you use node versioning.
- How many nodes you include in the npm package.

n8n recommends using the [n8n-node tool](#) to create the expected node file structure. You can customize the generated scaffolding as required to meet more complex needs.

Required files and directories

Your node must include:

- A `package.json` file at the root of the project. Every npm module requires this.
- A `nodes` directory, containing the code for your node:
 - This directory must contain the [base file](#), in the format `<node-name>.node.ts`. For example, `MyNode.node.ts`.
 - n8n recommends including a [codex file](#), containing metadata for your node. The codex filename must match the node base filename. For example, given a node base file named `MyNode.node.ts`, the codex name is `MyNode.node.json`.
 - The `nodes` directory can contain other files and subdirectories, including directories for versions, and node code split across more than one file to create a modular structure.
- A `credentials` directory, containing your credentials code. This code lives in a single [credentials file](#). The filename format is `<node-name>.credentials.ts`. For example, `MyNode.credentials.ts`.

Modular structure

You can choose whether to place all your node's functionality in one file, or split it out into a base file and other modules, which the base file then imports. Unless your node is very simple, it's a best practice to split it out.

A basic pattern is to separate out operations. Refer to the [HttpBin starter node](#) for an example of this.

For more complex nodes, n8n recommends a directory structure. Refer to the [Airtable node](#) or [Microsoft Outlook node](#) as examples.

- **actions**: a directory containing sub-directories that represent resources.
 - Each sub-directory should contain two types of files:
 - An index file with resource description (named either `<resourceName>.resource.ts` or `index.ts`)
 - Files for operations `<operationName>.operation.ts`. These files should have two exports: description of the operation and an execute function.
- **methods**: an optional directory dynamic parameters' functions.
- **transport**: a directory containing the communication implementation.

Versioning

If your node has more than one version, and you're using full versioning, this makes the file structure more complex. You need a directory for each version, along with a base file that sets the default version. Refer to [Node versioning](#) for more information on working with versions, including types of versioning.

Decide how many nodes to include in a package

There are two possible setups when building a node:

- One node in one npm package.
- More than one node in a single npm package.

n8n supports both approaches. If you include more than one node, each node should have its own directory in the `nodes` directory.

A best-practice example for programmatic nodes

n8n's built-in [Airtable node](#) implements a modular structure and versioning, following recommended patterns.

Node base file

The node base file contains the core code of your node. All nodes must have a base file. The contents of this file are different depending on whether you're building a declarative-style or programmatic-style node. For guidance on which style to use, refer to [Choose your node building approach](#).

These documents give short code snippets to help understand the code structure and concepts. For full walk-throughs of building a node, including real-world code examples, refer to [Build a declarative-style node](#) or [Build a programmatic-style node](#).

You can also explore the [n8n-nodes-starter](#) and n8n's own [nodes](#) for a wider range of examples. The starter contains basic examples that you can build on. The n8n [Mattermost node](#) is a good example of a more

complex programmatic-style node, including versioning.

For all nodes, refer to the:

- [Structure of the node base file](#)
- [Standard parameters](#)

For declarative-style nodes, refer to the:

- [Declarative-style parameters](#)

For programmatic-style nodes, refer to the:

- [Programmatic-style parameters](#)
 - [Programmatic-style execute\(\) method](#)
-

Structure of the node base file

The node base file follows this basic structure:

1. Add import statements.
2. Create a class for the node.
3. Within the node class, create a `description` object, which defines the node.

A programmatic-style node also has an `execute()` method, which reads incoming data and parameters, then builds a request. The declarative style handles this using the `routing` key in the `properties` object, within `descriptions`.

Outline structure for a declarative-style node

This code snippet gives an outline of the node structure.

```
import { INodeType, INodeTypeDescription } from 'n8n-workflow';

export class ExampleNode implements INodeType {
    description: INodeTypeDescription = {
        // Basic node details here
        properties: [
            // Resources and operations here
        ]
    };
}
```

Refer to [Standard parameters](#) for information on parameters available to all node types. Refer to [Declarative-style parameters](#) for the parameters available for declarative-style nodes.

Outline structure for a programmatic-style node

This code snippet gives an outline of the node structure.

```

import { IExecuteFunctions } from 'n8n-core';
import { INodeExecutionData, INodeType, INodeTypeDescription } from
'n8n-workflow';

export class ExampleNode implements INodeType {
    description: INodeTypeDescription = {
        // Basic node details here
        properties: [
            // Resources and operations here
        ]
    };

    async execute(this: IExecuteFunctions):
Promise<INodeExecutionData[][]> {
    // Process data and return
}
};


```

Refer to [Standard parameters](#) for information on parameters available to all node types. Refer to [Programmatic-style parameters](#) and [Programmatic-style execute method](#) for more information on working with programmatic-style nodes.

Standard parameters

These are the standard parameters for the [node base file](#). They're the same for all node types.

displayName

String | Required

This is the name users see in the n8n GUI.

name

String | Required

The internal name of the object. Used to reference it from other places in the node.

icon

String or Object | Required

Specifies an icon for a particular node. n8n recommends uploading your own image file.

You can provide the icon file name as a string, or as an object to handle different icons for light and dark modes. If the icon works in both light and dark modes, use a string that starts with `file:`, indicating the path to the icon file. For example:

```
icon: 'file:exampleNodeIcon.svg'
```

To provide different icons for light and dark modes, use an object with `light` and `dark` properties. For example:

```
icon: {
  light: 'file:exampleNodeIcon.svg',
  dark: 'file:exampleNodeIcon.dark.svg'
}
-8<- "_snippets/integrations/creating-nodes/node-icons.md"
```

group

Array of strings | Required

Tells n8n how the node behaves when the workflow runs. Options are:

- `trigger`: node waits for a trigger.
- `schedule`: node waits for a timer to expire.
- `input`, `output`, `transform`: these currently have no effect.
- An empty array, `[]`. Use this as the default option if you don't need trigger or schedule.

description

String | Required

A short description of the node. n8n uses this in the GUI.

defaults

Object | Required

Contains essential brand and name settings.

The object can include:

- `name`: String. Used as the node name on the canvas if the `displayName` is too long.
- `color`: String. Hex color code. Provide the brand color of the integration for use in n8n.

forceInputNodeExecution

Boolean | Optional

When building a multi-input node, you can choose to force all preceding nodes on all branches to execute before the node runs. The default is `false` (requiring only one input branch to run).

inputs

Array of strings | Required

Names the input connectors. Controls the number of connectors the node has on the input side. If you need only one connector, use `input: ['main']`.

outputs

Array of strings | Required

Names the output connectors. Controls the number of connectors the node has on the output side. If you need only one connector, use `output: ['main']`.

requiredInputs

Integer or Array | Optional

Used for multi-input nodes. Specify inputs by number that must have data (their branches must run) before the node can execute.

credentials

Array of objects | Required

This parameter tells n8n the credential options. Each object defines an authentication type.

The object must include:

- `name`: the credential name. Must match the `name` property in the credential file. For example, `name: 'asanaApi'` in `Asana.node.ts` links to `name = 'asanaApi'` in `AsanaApi.credential.ts`.
- `required`: Boolean. Specify whether authentication is required to use this node.

requestDefaults

Object | Required

Set up the basic information for the API calls the node makes.

This object must include:

- `baseURL`: The API base URL.

You can also add:

- `headers`: an object describing the API call headers, such as content type.
- `url`: string. Appended to the `baseURL`. You can usually leave this out. It's more common to provide this in the operations.

properties

Array of objects | Required

This contains the resource and operations objects that define node behaviors, as well as objects to set up mandatory and optional fields that can receive user input.

Resource objects

A resource object includes the following parameters:

- `displayName`: String. This should always be `Resource`.
- `name`: String. This should always be `resource`.
- `type`: String. Tells n8n which UI element to use, and what input type to expect. For example, `options` results in n8n adding a dropdown that allows users to choose one option. Refer to [Node UI elements](#) for more information.
- `noDataExpression`: Boolean. Prevents using an expression for the parameter. Must always be `true` for resource.

Operations objects

The operations object defines the available operations on a resource.

- `displayName`: String. This should always be `Options`.
- `name`: String. This should always be `option`.
- `type`: String. Tells n8n which UI element to use, and what input type to expect. For example, `dateTime` results in n8n adding a date picker. Refer to [Node UI elements](#) for more information.
- `noDataExpression`: Boolean. Prevents using an expression for the parameter. Must always be `true` for operation.
- `options`: Array of objects. Each objects describes an operation's behavior, such as its routing, the REST verb it uses, and so on. An `options` object includes:
 - `name`: String.
 - `value`: String.
 - `action`: String. This parameter combines the resource and operation. You should always include it, as n8n will use it in future versions. For example, given a resource called "Card" and an operation "Get all", your action is "Get all cards".
 - `description`: String.
 - `routing`: Object containing request details.

Additional fields objects

These objects define optional parameters. n8n displays them under **Additional Fields** in the GUI. Users can choose which parameters to set.

The objects must include:

```
displayName: 'Additional Fields',
name: 'additionalFields',
// The UI element type
type: '',
placeholder: 'Add Field',
default: {},
displayOptions: {
    // Set which resources and operations this field is available for
    show: [
        resource: [
            // Resource names
```

```
],
  operation: [
    // Operation names
  ]
},
}
```

For more information about UI element types, refer to [UI elements](#).

Declarative-style parameters

These are the parameters available for [node base file](#) of declarative-style nodes.

This document gives short code snippets to help understand the code structure and concepts. For a full walk-through of building a node, including real-world code examples, refer to [Build a declarative-style node](#).

Refer to [Standard parameters](#) for parameters available to all nodes.

methods and loadOptions

Object | Optional

methods contains the loadOptions object. You can use loadOptions to query the service to get user-specific settings, then return them and render them in the GUI so the user can include them in subsequent queries. The object must include routing information for how to query the service, and output settings that define how to handle the returned options. For example:

```
methods : {
  loadOptions: {
    routing: {
      request: {
        url: '/webhook/example-option-parameters',
        method: 'GET',
      },
      output: {
        postReceive: [
          {
            // When the returned data is nested under
            // another property
            type: 'rootProperty',
            properties: {
              property: 'responseData',
            },
          },
          {
            type: 'setKeyValue',
            properties: {
              name: '{$responseItem.key}',
              value: '{$responseItem.value}',
            },
          },
        ],
      },
    },
  },
}
```

```

        },
        {
          // If incoming data is an array of objects,
          // sort alphabetically by key
          type: 'sort',
          properties: {
            key: 'name',
          },
        },
      ],
    },
  ],
},

```

routing

Object | Required

routing is an object used within an options array in operations and input field objects. It contains the details of an API call.

The code example below comes from the [Declarative-style tutorial](#). It sets up an integration with a NASA API. It shows how to use requestDefaults to set up the basic API call details, and routing to add information for each operation.

```

description: INodeTypeDescription = {
  // Other node info here
  requestDefaults: {
    baseURL: 'https://api.nasa.gov',
    url: '',
    headers: {
      Accept: 'application/json',
      'Content-Type': 'application/json',
    },
  },
  properties: [
    // Resources here
    {
      displayName: 'Operation'
      // Other operation details
      options: [
        {
          name: 'Get'
          value: 'get',
          description: '',
          routing: {
            request: {
              method: 'GET',
              url: '/planetary/apod'
            }
          }
        }
      ]
    }
  ]
}

```

version

Number or Array | Optional

If you have one version of your node, this can be a number. If you want to support more than one version, turn this into an array, containing numbers for each node version.

n8n supports two methods of node versioning, but declarative-style nodes must use the light versioning approach. Refer to [Node versioning](#) for more information.

Programmatic-style parameters

These are the parameters available for [node base file](#) of programmatic-style nodes.

This document gives short code snippets to help understand the code structure and concepts. For a full walk-through of building a node, including real-world code examples, refer to [Build a programmatic-style node](#).

Programmatic-style nodes also use the `execute()` method. Refer to [Programmatic-style execute method](#) for more information.

Refer to [Standard parameters](#) for parameters available to all nodes.

defaultVersion

Number | Optional

Use `defaultVersion` when using the full versioning approach.

n8n support two methods of node versioning. Refer to [Node versioning](#) for more information.

methods and loadOptions

Object | Optional

Contains the `loadOptions` method for programmatic-style nodes. You can use this method to query the service to get user-specific settings (such as getting a user's email labels from Gmail), then return them and render them in the GUI so the user can include them in subsequent queries.

For example, n8n's [Gmail node](#) uses `loadOptions` to get all email labels:

```
methods = {
    loadOptions: {
        // Get all the labels and display them
        async getLabels(
            this: ILoadOptionsFunctions,
        ): Promise<INodePropertyOptions[] > {
            const returnData: INodePropertyOptions[] = [];
            const labels = await googleApiRequestAllItems.call(
```

```

        this,
        'labels',
        'GET',
        '/gmail/v1/users/me/labels',
    );
    for (const label of labels) {
        const labelName = label.name;
        const labelId = label.id;
        returnData.push({
            name: labelName,
            value: labelId,
        });
    }
    return returnData;
},
},
);

```

version

Number or Array | Optional

Use `version` when using the light versioning approach.

If you have one version of your node, this can be a number. If you want to support multiple versions, turn this into an array, containing numbers for each node version.

n8n support two methods of node versioning. Programmatic-style nodes can use either. Refer to [Node versioning](#) for more information.

Programmatic-style execute() method

The main difference between the declarative and programmatic styles is how they handle incoming data and build API requests. The programmatic style requires an `execute()` method, which reads incoming data and parameters, then builds a request. The declarative style handles requests using the `routing key` in the `operations` object.

The `execute()` method creates and returns an instance of `INodeExecutionData`.

Node codex files

The codex file contains metadata about your node. This file is the JSON file at the root of your node. For example, the `HttpBin.node.json` file in the n8n starter.

The codex filename must match the node base filename. For example, given a node base file named `MyNode.node.ts`, the codex would be named `MyNode.node.json`.

| Parameter | Description |
|--------------|---|
| node | Includes the node name. Must start with n8n-nodes-base.. For example, n8n-nodes-base.openweathermap. |
| nodeVersion | The node version. This should have the same value as the version parameter in your main node file. For example, "1.0". |
| codexVersion | The codex file version. The current version is "1.0". |
| categories | The settings in the categories array determine which category n8n adds your node to in the GUI. See Node categories for more information. |
| resources | The resources object contains links to your node documentation. n8n automatically adds help links to credentials and nodes in the GUI. |

Node categories

You can define one or more categories in your node configuration JSON. This helps n8n put the node in the correct category in the nodes panel.

Choose from these categories:

- Data & Storage
- Finance & Accounting
- Marketing & Content
- Productivity
- Miscellaneous
- Sales
- Development
- Analytics
- Communication
- Utility

You must match the syntax. For example, Data & Storage not data and storage.

Credentials file

The credentials file defines the authorization methods for the node. The settings in this file affect what n8n displays in the **Credentials** modal, and must reflect the authentication requirements of the service you're connecting to.

In the credentials file, you can use all the [n8n UI elements](#). n8n encrypts the data that's stored using credentials using an encryption key.

Structure of the credentials file

The credentials file follows this basic structure:

1. Import statements
2. Create a class for the credentials
3. Within the class, define the properties that control authentication for the node.

Outline structure

```
import {
    IAuthenticateGeneric,
    ICredentialTestRequest,
    ICredentialType,
    INodeProperties,
} from 'n8n-workflow';

export class ExampleNode implements ICredentialType {
    name = 'exampleNodeApi';
    displayName = 'Example Node API';
    documentationUrl = '';
    properties: INodeProperties[] = [
        {
            displayName: 'API Key',
            name: 'apiKey',
            type: 'string',
            default: '',
        },
    ];
    authenticate: IAuthenticateGeneric = {
        type: 'generic',
        properties: {
            // Can be body, header, qs or auth
            qs: {
                // Use the value from `apiKey` above
                'api_key': `=${{$credentials.apiKey}}`
            }
        },
    };
    test: ICredentialTestRequest = {
        request: {
            baseURL: `=${{$credentials?.domain}}`,
            url: '/bearer',
        },
    };
}
```

Parameters

name

String. The internal name of the object. Used to reference it from other places in the node.

displayName

String. The name n8n uses in the GUI.

documentationUrl

String. URL to your credentials documentation.

properties

Each object contains:

- `displayName`: the name n8n uses in the GUI.
- `name`: the internal name of the object. Used to reference it from other places in the node.
- `type`: the data type expected, such as `string`.
- `default`: the URL that n8n should use to test credentials.

authenticate

- `authenticate`: Object. Contains objects that tell n8n how to inject the authentication data as part of the API request.

type

String. If you're using an authentication method that sends data in the header, body, or query string, set this to 'generic'.

properties

Object. Defines the authentication methods. Options are:

- `body`: Object. Sends authentication data in the request body. Can contain nested objects.

```
authenticate: IAuthenticateGeneric = {
    type: 'generic',
    properties: {
        body: {
            username: `={{credentials.username}}`,
            password: `={{credentials.password}}`,
        },
    },
};
```

- `header`: Object. Send authentication data in the request header.

```
authenticate: IAuthenticateGeneric = {
    type: 'generic',
    properties: {
        header: {
            Authorization: `=Bearer {{$credentials.authToken}}`,
        },
    },
};
```

- `qs`: Object. Stands for "query string." Send authentication data in the request query string.

```
authenticate: IAuthenticateGeneric = {
    type: 'generic',
    properties: {
        qs: {

```

```
        token: '={`${$credentials.token}`}',  
    },  
},  
};
```

- auth: Object. Used for Basic Auth. Requires username and password as the key names.

```
authenticate: IAuthenticateGeneric = {  
    type: 'generic',  
    properties: {  
        auth: {  
            username: '`${$credentials.username}`',  
            password: '`${$credentials.password}`',  
        },  
    },  
};
```

test

Provide a request object containing a URL and authentication type that n8n can use to test the credential.

```
test: ICredentialTestRequest = {  
    request: {  
        baseURL: '`${$credentials?.domain}`',  
        url: '/bearer',  
    },  
};
```

HTTP request helper for node builders

n8n provides a flexible helper for making HTTP requests, which abstracts away most of the complexity.

Usage

Call the helper inside the execute function.

```
// If no auth needed  
const response = await this.helpers.httpRequest(options);  
  
// If auth needed  
const response = await  
this.helpers.httpRequestWithAuthentication.call(  
    this,  
    'credentialTypeName', // For example: pipedriveApi  
    options,  
);
```

options is an object:

```
{  
    url: string;  
    headers?: object;
```

```

        method?: 'GET' | 'POST' | 'PUT' | 'DELETE' | 'HEAD';
        body?: FormData | Array | string | number | object | Buffer |
URLSearchParams;
        qs?: object;
        arrayFormat?: 'indices' | 'brackets' | 'repeat' | 'comma';
        auth?: {
            username: string,
            password: string,
        };
        disableFollowRedirect?: boolean;
        encoding?: 'arraybuffer' | 'blob' | 'document' | 'json' | 'text'
    | 'stream';
        skipSslCertificateValidation?: boolean;
        returnFullResponse?: boolean;
        proxy?: {
            host: string;
            port: string | number;
            auth?: {
                username: string;
                password: string;
            },
            protocol?: string;
        };
        timeout?: number;
        json?: boolean;
    }
}

```

`url` is required. The other fields are optional. The default method is `GET`.

Some notes about the possible fields:

- `body`: you can use a regular JavaScript object for JSON payload, a buffer for file uploads, an instance of `FormData` for `multipart/form-data`, and `URLSearchParams` for `application/x-www-form-urlencoded`.
- `headers`: a key-value pair.
 - If `body` is an instance of `FormData` then n8n adds `content-type: multipart/form-data` automatically.
 - If `body` is an instance of `URLSearchParams`, then n8n adds `content-type: application/x-www-form-urlencoded`.
 - To override this behavior, set a `content-type` header.
- `arrayFormat`: if your query string contains an array of data, such as `const qs = {IDs: [15,17]}`, the value of `arrayFormat` defines how n8n formats it.
 - `indices` (default): { `a: ['b', 'c']` } as `a[0]=b&a[1]=c`
 - `brackets`: { `a: ['b', 'c']` } as `a[]>=b&a[]>=c`
 - `repeat`: { `a: ['b', 'c']` } as `a=b&a=c`
 - `comma`: { `a: ['b', 'c']` } as `a=b,c`
- `auth`: Used for Basic auth. Provide `username` and `password`. n8n recommends omitting this, and using `helpers.httpRequestWithAuthentication(...)` instead.
- `disableFollowRedirect`: By default, n8n follows redirects. You can set this to true to prevent this from happening.
- `skipSslCertificateValidation`: Used for calling HTTPS services without proper certificate
- `returnFullResponse`: Instead of returning just the body, returns an object with more data in the following format: {`body: body, headers: object, statusCode: 200, statusMessage: 'OK'`}
- `encoding`: n8n can detect the content type, but you can specify `arrayBuffer` to receive a Buffer you can read from and interact with.

Example

For an example, refer to the [Mattermost node](#).

Deprecation of the previous helper

The previous helper implementation using `this.helpers.request(options)` used and exposed the `request-promise` library. This was removed in version 1.

To minimize incompatibility, n8n made a transparent conversion to another library called Axios.

If you are having issues, please report them in the [Community Forums](#) or on [GitHub](#).

Migration guide to the new helper

The new helper is much more robust, library agnostic, and easier to use.

New nodes should all use the new helper. You should strongly consider migrating existing custom nodes to the new helper. These are the main considerations when migrating:

- Accepts `url`. Doesn't accept `uri`.
 - `encoding: null` now must be `encoding: arrayBuffer`.
 - `rejectUnauthorized: false` is now `skipSslCertificateValidation: true`
 - Use `body` according to `content-type` headers to clarify the payload.
 - `resolveWithFullResponse` is now `returnFullResponse` and has similar behavior
-

Item linking

-8<- “_snippets/data/data-mapping/item-linking-node-creators.md”

UX guidelines for community nodes

Your node's UI must conform to these guidelines to be a [verified community node candidate](#).

Credentials

API key and sensitive credentials should always be password fields.

OAuth

Always include the OAuth credential if available.

Node structure

Operations to include

Try to include **CRUD** operations for each resource type.

Try to include common operations in nodes for each resource. n8n uses some CRUD operations to keep the experience consistent and allow users to perform basic operations on the resource. The suggested operations are:

- **Create**
- **Create or Update (Upsert)**
- **Delete**
- **Get**
- **Get Many:** also used when some filtering or search is available
- **Update**

Notes:

1. These operations can apply to the resource itself or an entity inside of the resource (for example, a row inside a Google Sheet). When operating on an entity inside of the resource, you must specify the **name of the entity** in the operations name.
2. The naming could change depending on the node and the resource. Check the following guidelines for details.

Resource Locator

- Use a Resource Locator component whenever possible. This provides a much better UX for users. The Resource Locator Component is most often useful when you have to select a single item.
- The default option for the Resource Locator Component should be `From list` (if available).

Consistency with other nodes

- Maintain UX consistency: n8n tries to keep its UX consistent. This means following existing UX patterns, in particular, those used in the latest new or overhauled nodes.
- Check similar nodes: For example, if you're working on a database node, it's worth checking the Postgres node.

Sorting options

- You can enhance certain "Get Many" operations by providing users with sorting options.
- Add sorting in a dedicated collection (below the "Options" collection). Follow the example of [Airtable Record:Search](#).

Node functionality

Deleting operations output

When deleting an item (like a record or a row), return an array with a single object: {"deleted": true}. This is a confirmation for the user that the deletion was successful and the item will trigger the following node.

Simplifying output fields

Normal nodes: 'Simplify' parameter

When an endpoint returns data with more than 10 fields, add the "Simplify" boolean parameter to return a simplified version of the output with max 10 fields.

- One of the main issues with n8n can be the size of data and the Simplify parameter limits that problem by reducing data size.
- Select the most useful fields to output in the simplified node and sort them to have the most used ones at the top.
- In the Simplify mode, it's often best to flatten nested fields
- Display Name: Simplify
- Description: Whether to return a simplified version of the response instead of the raw data

AI tool nodes: 'Output' parameter

When an endpoint returns data with more than 10 fields, add the 'Output' option parameter with 3 modes.

In AI tool nodes, allow the user to be more granular and select the fields to output. The rationale is that tools may run out of context window and they can get confused by too many fields, so it's better to pass only the ones they need.

Options:

- **Simplified:** Works the same as the "Simplify" parameter described above.
- **Raw:** Returns all the available fields.
- **Selected fields:** Shows a multi-option parameter for selecting the fields to add to the output and send to the AI agent. By default, this option always returns the ID of the record/entity.

Copy

Text Case

Use **Title Case** for the node name, parameters display names (labels), dropdown titles. Title Case is when you capitalize the first letter of each word, except for certain small words, such as articles and short prepositions.

Use **Sentence case** for node action names, node descriptions, parameters descriptions (tooltips), hints, dropdown descriptions.

Terminology

- **Use the third-party service terminology:** Try to use the same terminology as the service you're interfacing with (for example, Notion 'blocks', not Notion 'paragraphs').
- **Use the terminology used in the UI:** Stick to the terminology used in the user interface of the service, rather than that used in the APIs or technical documentation (for example, in Trello you "archive" cards, but in the API they show up as "closed". In this case, you might want to use "archive").
- **No tech jargon:** Don't use technical jargon where simple words will do. For example, use "field" instead of "key".
- **Consistent naming:** Choose one term for something and stick to it. For example, don't mix "directory" and "folder".

Placeholders

It's often helpful to insert examples of content in parameters placeholders. These should start with "e.g." and use **camel case** for the demo content in fields.

Placeholder examples to copy:

- image: e.g. <https://example.com/image.png>
- video: e.g. <https://example.com/video.mp4>
- search term: e.g. automation
- email: e.g. nathan@example.com
- Twitter user (or similar): e.g. n8n
- Name and last name: e.g. Nathan Smith
- First name: e.g. Nathan
- Last name: e.g. Smith

Operations name, action, and description

- **Name:** This is the name displayed in the select when the node is open on the canvas. It must use title case and doesn't have to include the resource (for example, "Delete").
- **Action:** This is the name of the operation displayed in the panel where the user selects the node. It must be in sentence case and must include the resource (for example, "Delete record").
- **Description:** This is the sub-text displayed below the name in the select when the node is open on the canvas. It must use sentence case and must include the resource. It can add a bit of information and use alternative words than the basic resource/operation (for example, "Retrieve a list of users").
- If the operation acts on an entity that's not the Resource (for example, a row in a Google Sheet), specify that in the operation name (for example, "Delete Row").

As a general rule, is important to understand what the **object** of an operation is. Sometimes, the object of an Operation is the resource itself (for example, Sheet:Delete to delete a Sheet).

In other cases, the object of the operation isn't the resource, but something contained inside the resource (for example, Table:Delete rows, here the resource is the table, but what you are operating on are the rows inside of it).

Naming name

This is the name displayed in the select when the node is open on the canvas.

- Parameter: name
- Case: Title Case

Naming guidelines:

- **Don't repeat the resource (if the resource selection is above):** The resource is often displayed above the operation, so it's not necessary to repeat it in the operation (this is the case if the object of the operation is the resource itself).
 - For example: Sheet:Delete → No need to repeat Sheet in Delete, because n8n displays Sheet in the field above and what you're deleting is the Sheet.
- **Specify the resource if there's no resource selection above:** In some nodes, you won't have a resource selection (because there's only one resource). In these cases, specify the resource in the operation.
 - For example: Delete Records → In Airtable, there's no resource selection, so it's better to specify that the Delete operation will delete records.
- **Specify the object of the operation if it's not the resource:** Sometimes, the object of the operation isn't the resource. In these cases, specify the object in the operation as well.
 - For example: Table:Get Columns → Specify Columns because the resource is Table, while the object of the operation is Columns.

Naming action

This is the name of the operation displayed in the panel where the user selects the node. * Parameter: action * Case: Sentence case

Naming guidelines:

- **Omit articles:** To keep the text shorter, get rid of articles (a, an, the...).
 - **correct:** Update row in sheet
 - **incorrect:** Update a row in a sheet
- **Repeat the resource:** In this case, it's okay to repeat the resource. Even if the resource is visible in the list, the user might not notice and it's useful to repeat it in the operation label.
- **Specify the object of the operation if it is not the resource:** Same as for the operation name. In this case, you don't need to repeat the resource.
 - For example: Append Rows → You have to specify Rows because rows are what you're actually appending to. Don't add the resource (Sheet) since you aren't appending to the resource.

Naming description

This is the subtext displayed below the name in the selection when the node is open on the canvas.

- Parameter: description
- Case: Sentence case

Naming guidelines:

- If possible, add more information than that specified in the

operation name

- Use alternative wording to help users better understand what the operation is doing. Some people might not understand the text used in the operation (maybe English isn't their native language), and using alternative working could help them.

Vocabulary

n8n uses a general vocabulary and some context-specific vocabulary for groups of similar applications (for example, databases or spreadsheets).

The general vocabulary takes inspiration from CRUD operations:

- **Clear**

- Delete all the contents of the resource (empty the resource).
- Description: Delete all the <CHILD_ELEMENT>s inside the <RESOURCE>

- **Create**

- Create a new instance of the resource.
- Description: Create a new <RESOURCE>

- **Create or Update**

- Create or update an existing instance of the resource.
- Description: Create a new <RESOURCE> or update an existing one (upsert)

- **Delete**

- You can use "Delete" in two different ways:
 1. Delete a resource:
 - Description: Delete a <RESOURCE> permanently (use "permanently" only if that's the case)
 2. Delete something **inside** of the resource (for example, a row):
 - In this case, **always specify the object of the operation**: for example, Delete Rows or Delete Records.
 - Description: Delete a <CHILD_ELEMENT> permanently

- **Get**

- You can use "Get" in two different ways:
 1. Get a resource:
 - Description: Retrieve a <RESOURCE>
 2. Get an item **inside** of the resource (for example, records):
 - In this case, **always specify the object of the operation**: for example, Get Row or Get Record.
 - Description: Retrieve a <CHILD_ELEMENT> from the/a <RESOURCE>

- **Get Many**

- You can use "Get Many" in two different ways:
 1. Get a list of resources (without filtering):
 - Description: Retrieve a list of <RESOURCE>s
 2. Get a list of items **inside** of the resource (for example, records):
 - In this case, **always specify the object of the operation**: for example, Get Many Rows or Get Many Records.
 - You can omit Many: Get Many Rows can be Get Rows.
 - Description: List all <CHILD_ELEMENT>s in the/a <RESOURCE>

- **Insert or Append**

- Add something inside of a resource.
- Use insert for database nodes.

- Description: Insert <CHILD_ELEMENT>(s) in a <RESOURCE>
- **Insert or Update or Append or Update**
 - Add or update something inside of a resource.
 - Use insert for database nodes.
 - Description: Insert <CHILD_ELEMENT>(s) or update an existing one(s) (upsert)
- **Update**
 - You can use “Update” in two different ways:
 1. Update a resource:
 - Description: Update one or more <RESOURCE>s
 2. Update something **inside** of a resource (for example, a row):
 - In this case, **always specify the object of the operation**: for example, Update Rows or Update Records.
 - Description: Update <CHILD_ELEMENT>(s) inside a <RESOURCE>

Referring to parameter and field name

When you need to refer to parameter names or field names in copy, wrap them in single quotation marks (for example, “Please fill the ‘name’ parameter).

Boolean description

Start the description of boolean components with ‘Whether...’

Errors

General philosophy

Errors are sources of pain for users. For this reason, n8n always wants to tell the user:

- **What happened:** a description of the error and what went wrong.
- **How to solve the problem:** or at least how to get unstuck and continue using n8n without problems. n8n doesn’t want users to remain blocked, so use this as an opportunity to guide them to success.

Error structure in the Output panel

Error Message - What happened

This message explains to the user what happened, and the current issue that prevents the execution completing.

- If you have the displayName of the parameter that triggered the error, include it in the error message or description (or both).
- Item index: if you have the ID of the item that triggered the error, append [Item X] to the error message. For example, The ID of the release in the parameter “Release ID” for could not be found [item 2].
- Avoid using words like “error”, “problem”, “failure”, “mistake”.

Error Description - How to solve or get unstuck

The description explains to users how to solve the problem, what to change in the node configuration (if that's the case), or how to get unstuck. Here, you should guide them to the next step and unblock them.

Avoid using words like "error", "problem", "failure", "mistake".

Community node verification guidelines

Use the n8n-node tool

All verified community node authors should strongly consider using the [n8n-node tool](#) to create and check their package. This helps n8n ensure quality and consistency by:

- Generating the expected package file structure
- Adding the required metadata and configuration to the `package.json` file
- Making it easy to lint your code against n8n's standards
- Allowing you to load your node in a local n8n instance for testing

Package source verification

- Verify that your npm package repository URL matches the expected GitHub (or other platform) repository.
- Confirm that the package author / maintainer matches between npm and the repository.
- Confirm that the git link in npm works and that the repository is public.
- Make sure your package has proper documentation (README, usage examples, etc.).
- Make sure your package license is MIT.

No external dependencies

- Ensure that your package does **not** include any external dependencies to keep it lightweight and easy to maintain.

Proper documentation

- Provide clear documentation, whether it's a **README** on GitHub or links to relevant **API documentation**.
- Include usage instructions, example workflows, and any necessary authentication details.

No access to environment variables or file system

- The code **must not** interact with environment variables or attempt

- to read/write files.
- Pass all necessary data through node parameters.

Follow n8n best practices

- Maintain a clear and consistent coding style.
- Use **TypeScript** and follow n8n's [node development guidelines](#).
- Ensure proper error handling and validation.
- Make sure the linter passes (in other words, make sure running `npx @n8n/scan-community-package n8n-nodes-PACKAGE` passes).

Use English language only

- Both the node interface and all documentation must be in **English** only.
 - This includes parameter names, descriptions, help text, error messages and **README** content.
-

Test a node

This section contains information about testing your node.

There are two ways to test your node:

- Manually, by [running it on your own machine](#) within a local n8n instance.
- Automatically, using the [linter](#).

You should use both methods before publishing your node.

Run your node locally

-8<- “_snippets/integrations/creating-nodes/testing.md”

n8n node linter

n8n's node linter, [eslint-plugin-n8n-nodes-base](#), statically analyzes (“lints”) the source code of n8n nodes and credentials in the official repository and in community packages. The linter detects issues and automatically fixes them to help you follow best practices.

`eslint-plugin-n8n-nodes-base` contains a [collection of rules](#) for node files (`*.node.ts`), resource description files (`*Description.ts`), credential files (`*.credentials.ts`), and the `package.json` of a community package.

Setup

If using the [n8n node starter](#): Run `npm install` in the starter project to install all dependencies. Once the installation finishes, the linter is available to you.

If using VS Code, install the [ESLint VS Code extension](#). For other IDEs, refer to their ESLint integrations.

Usage

You can use the linter in a community package or in the main n8n repository.

Linting

In a community package, the linter runs automatically after installing dependencies and before publishing the package to npm. In the [main n8n repository](#), the linter runs automatically using GitHub Actions whenever you push to your pull request.

In both cases, VS Code lints in the background as you work on your project. Hover over a detected issue to see a full description of the linting and a link to further information.

You can also run the linter manually:

- Run `npm run lint` to lint and view detected issues in your console.
- Run `npm run lintfix` to lint and automatically fix issues. The linter fixes violations of rules [marked as automatically fixable](#).

Both commands can run in the root directory of your community package, or in `/packages/nodes-base/` in the main repository.

Exceptions

Instead of fixing a rule violation, you can also make an exception for it, so the linter doesn't flag it.

To make a lint exception from VS Code: hover over the issue and click on Quick fix (or cmd+ . in macOS) and select **Disable {rule} for this line**. Only disable rules for a line where you have good reason to. If you think the linter is incorrectly reporting an issue, please [report it in the linter repository](#).

To add a lint exception to a single file, add a code comment. In particular, TSLint rules may not show up in VS Code and may need to be turned off using code comments. Refer to the [TSLint documentation](#) for more guidance.

Troubleshooting

Credentials

Error message: 'Credentials of type “*” aren’t known’

Check that the name in the credentials array matches the name used in the property name of the credentials' class.

[Image: Troubleshooting credentials]

Editor UI

Error message: 'There was a problem loading init data: API-Server can not be reached. It's probably down'

- Check that the names of the node file, node folder, and class match the path added to packages/nodes-base/package.json.
- Check that the names used in the displayOptions property are names used by UI elements in the node.

Node icon doesn't show up in the Add Node menu and the Editor UI

- Check that the icon is in the same folder as the node.
- Check that it's either in PNG or SVG format.
- When the icon property references the icon file, check that it includes the logo extension (.png or .svg) and that it prefixes it with file:. For example, file:friendGrid.png or file:friendGrid.svg.

Node icon doesn't fit

- If you use an SVG file, make sure the canvas size is square. You can find instructions to change the canvas size of an SVG file using GIMP [here](#).
- If you use a PNG file, make sure that it's 60x60 pixels.

Node doesn't show up in the Add Node menu

Check that you registered the node in the package.json file in your project.

Changes to the description properties don't show in the UI on refreshing

Every time you change the description properties, you have to stop the current n8n process (`ctrl + c`) and run it again. You may also need to re-run `npm link`.

Linter incorrectly warning about file name case

The node linter has rules for file names, including what case they should be. Windows users may encounter an issue when renaming files that causes the linter to continue giving warnings, even after you rename the files. This is due to a [known Windows issue](#) with changing case when renaming files.

Deploy a node

This section contains details on how to deploy and share your node.

You can choose to:

- Submit your node to the community node repository. This makes it available for everyone to use, and allows you to install and use it like any other community node. This is the only way to use custom nodes on cloud.
 - Install the node into your n8n instance as a private node.
-

Submit community nodes

-8<- “_snippets/integrations/submit-community-node.md”

Install private nodes

You can build your own nodes and install them in your n8n instance without publishing them on npm. This is useful for nodes that you create for internal use only at your company.

Install your node in a Docker n8n instance

If you’re running n8n using Docker, you need to create a Docker image with the node installed in n8n.

1. Create a Dockerfile and paste the code from this Dockerfile.

Your Dockerfile should look like this:

```
FROM node:16-alpine

ARG N8N_VERSION

RUN if [ -z "$N8N_VERSION" ] ; then echo "The N8N_VERSION argument is missing!" ; exit 1; fi

# Update everything and install needed dependencies
RUN apk add --update graphicsmagick tzdata git tini su-exec

# Set a custom user to not have n8n run as root
USER root

# Install n8n and the packages it needs to build it correctly.
RUN apk --update add --virtual build-dependencies python3 build-base ca-certificates && \
    npm config set python "$(which python3)" && \
    npm_config_user=root npm install -g full-icu
n8n@${N8N_VERSION} && \
    apk del build-dependencies \
&& rm -rf /root /tmp/* /var/cache/apk/* && mkdir /root;
```

```

# Install fonts
RUN apk --no-cache add --virtual fonts msttcorefonts-installer
fontconfig && \
    update-ms-fonts && \
    fc-cache -f && \
    apk del fonts && \
    find /usr/share/fonts/truetype/msttcorefonts/ -type l -exec
unlink {} \; \
&& rm -rf /root /tmp/* /var/cache/apk/* && mkdir /root

ENV NODE_ICU_DATA /usr/local/lib/node_modules/full-icu

WORKDIR /data

COPY docker-entrypoint.sh /docker-entrypoint.sh
ENTRYPOINT ["tini", "--", "/docker-entrypoint.sh"]

EXPOSE 5678/tcp

```

2. Compile your custom node code (`npm run build` if you are using nodes starter). Copy the **node** and **credential** folders from within the **dist** folder into your container's `~/.n8n/custom/` directory. This makes them available to Docker.
3. Download the [docker-entrypoint.sh](#) file, and place it in the same directory as your Dockerfile.
4. Build your Docker image:

```

# Replace <n8n-version-number> with the n8n release version
# number.
# For example, N8N_VERSION=0.177.0
docker build --build-arg N8N_VERSION=<n8n-version-number> --
tag=customizedn8n .

```

You can now use your node in Docker.

Install your node in a global n8n instance

If you've installed n8n globally, make sure that you install your node inside n8n. n8n will find the module and load it automatically.

Self-hosting n8n

This section provides guidance on setting up n8n for both the Enterprise and Community self-hosted editions. The Community edition is free, the Enterprise edition isn't.

See [Community edition features](#) for a list of available features.

- **Installation and server setups**

Install n8n on any platform using npm or Docker. Or follow our guides to popular hosting platforms.

[:octicons-arrow-right-24: Docker installation guide](#)

- **Configuration**

Learn how to configure n8n with environment variables.

[:octicons-arrow-right-24: Environment Variables](#)

- **Users and authentication**

Choose and set up user authentication for your n8n instance.

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

- **Scaling**

Manage data, modes, and processes to keep n8n running smoothly at scale.

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

- **Securing n8n**

Secure your n8n instance by setting up SSL, SSO, or 2FA or blocking or opting out of some data collection or features.

[:octicons-arrow-right-24: Securing n8n guide](#)

- **Starter kits**

New to n8n or AI? Try our Self-hosted AI Starter Kit. Curated by n8n, it combines the self-hosted n8n platform with compatible AI products and components to get you started building self-hosted AI workflows.

[:octicons-arrow-right-24: Starter kits](#)

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

Community Edition Features

The community edition includes almost the complete feature set of n8n, except for the features listed here.

The community edition doesn't include these features:

- [Custom Variables](#)
- [Environments](#)
- [External secrets](#)
- [External storage for binary data](#)
- [Log streaming \(Logging is included\)](#)
- [Multi-main mode \(Queue mode is included\)](#)
- [Projects](#)
- [SSO \(SAML, LDAP\)](#)
- [Sharing \(workflows, credentials\)](#) (Only the instance owner and the user who creates them can access workflows and credentials)
- [Version control using Git](#)

These features are available on the Enterprise Cloud plan, including the self-hosted Enterprise edition. Some of these features are available on the Starter and Pro Cloud plan.

See [pricing](#) for reference.

Registered Community Edition

You can unlock extra features by registering your n8n community edition. You register with your email and receive a license key.

Registering unlocks these features for the community edition:

- Folders: Organize your workflows into tidy folders
- [Debug in editor](#): Copy and [pin](#) execution data when working on a workflow
- [Custom execution data](#): Save, find, and annotate execution metadata

To register a new community edition instance, select the option during your initial account creation.

To register an existing community edition instance:

1. Select the **three dots icon** Image: three dots icon in the lower-left corner.
2. Select **Settings** and then **Usage and plan**.
3. Select **Unlock** to enter your email and then select **Send me a free license key**.
4. Check your email for the account you entered.

Once you have a license key, activate it by clicking the button in the license email or by visiting **Options > Settings > Usage and plan** and selecting **Enter activation key**.

Once activated, your license will not expire. We may change the unlocked features in the future. This will not impact previously unlocked features.

npm

npm is a quick way to get started with n8n on your local machine. You must have [Node.js](#) installed. n8n requires a Node.js version between 20.19 and 24.x, inclusive.

-8<- “_snippets/self-hosting/installation/latest-next-version.md”

Try n8n with npx

You can try n8n without installing it using npx.

From the terminal, run:

```
npx n8n
```

This command will download everything that's needed to start n8n. You can then access n8n and start building workflows by opening <http://localhost:5678>.

Install globally with npm

To install n8n globally, use npm:

```
npm install n8n -g
```

To install or update to a specific version of n8n use the @ syntax to specify the version. For example:

```
npm install -g n8n@0.126.1
```

To install next:

```
npm install -g n8n@next
```

After the installation, start n8n by running:

```
n8n  
# or  
n8n start
```

Next steps

Try out n8n using the [Quickstarts](#).

Updating

To update your n8n instance to the latest version, run:

```
npm update -g n8n
```

To install the next version:

```
npm install -g n8n@next  
-8<- "_snippets/self-hosting/installation/tunnel.md"
```

Start n8n with --tunnel by running:

```
n8n start --tunnel
```

Reverting an upgrade

Install the older version that you want to go back to.

If the upgrade involved a database migration:

1. Check the feature documentation and release notes to see if there are any manual changes you need to make.
2. Run `n8n db:revert` on your current version to roll back the database. If you want to revert more than one database migration, you need to repeat this process.

Windows troubleshooting

If you are experiencing issues running n8n on Windows, make sure your Node.js environment is correctly set up. Follow Microsoft's guide to [Install NodeJS on Windows](#).

Docker Installation

n8n recommends using [Docker](#) for most self-hosting needs. It provides a clean, isolated environment, avoids operating system and tooling incompatibilities, and makes database and environment management simpler.

You can also use n8n in Docker with [Docker Compose](#). You can find Docker Compose configurations for various architectures in the [n8n-hosting repository](#).

```
-8<- "_snippets/self-hosting/warning.md"
```

You can also follow along with our video guide here:

Prerequisites

Before proceeding, install Docker:

- [Docker Desktop](#) is available for Mac, Windows, and Linux. Docker Desktop includes the Docker Engine and Docker Compose.
- [Docker Engine](#) and [Docker Compose](#) are also available as separate packages for Linux. Use this for Linux machines without a graphical environment or when you don't want the Docker Desktop UI.

```
-8<- "_snippets/self-hosting/installation/latest-next-version.md"
```

Starting n8n

From your terminal, run the following commands, replacing the `<YOUR_TIMEZONE>` placeholders with [your timezone](#):

```
docker volume create n8n_data

docker run -it --rm \
--name n8n \
-p 5678:5678 \
-e GENERIC_TIMEZONE="<YOUR_TIMEZONE>" \
-e TZ="<YOUR_TIMEZONE>" \
-e N8N_ENFORCE_SETTINGS_FILE_PERMISSIONS=true \
-e N8N_RUNNERS_ENABLED=true \
-v n8n_data:/home/node/.n8n \
docker.n8n.io/n8nio/n8n
```

This command creates a volume to store persistent data, downloads the required n8n image, and starts the container with the following settings:

- Maps and exposes port 5678 on the host.
- Sets the timezone for the container:
 - the `TZ` environment variable sets the system timezone to control what scripts and commands like `date` return.
 - the `GENERIC_TIMEZONE` environment variable sets the correct timezone for schedule-oriented nodes like the [Schedule Trigger node](#).

- Enforces secure file permissions for the n8n configuration file.
- Enables task runners, the recommended way of executing tasks in n8n.
- Mounts the n8n_data volume to the /home/node/.n8n directory to persist your data across container restarts.

Once running, you can access n8n by opening: <http://localhost:5678>

Using with PostgreSQL

By default, n8n uses SQLite to save credentials, past executions, and workflows. n8n also supports PostgreSQL, configurable using environment variables as detailed below.

To use n8n with PostgreSQL, execute the following commands, replacing the placeholders (depicted within angled brackets, for example <POSTGRES_USER>) with your actual values:

```
docker volume create n8n_data

docker run -it --rm \
--name n8n \
-p 5678:5678 \
-e GENERIC_TIMEZONE=<YOUR_TIMEZONE> \
-e TZ=<YOUR_TIMEZONE> \
-e N8N_ENFORCE_SETTINGS_FILE_PERMISSIONS=true \
-e N8N_RUNNERS_ENABLED=true \
-e DB_TYPE=postgresdb \
-e DB_POSTGRESDB_DATABASE=<POSTGRES_DATABASE> \
-e DB_POSTGRESDB_HOST=<POSTGRES_HOST> \
-e DB_POSTGRESDB_PORT=<POSTGRES_PORT> \
-e DB_POSTGRESDB_USER=<POSTGRES_USER> \
-e DB_POSTGRESDB_SCHEMA=<POSTGRES_SCHEMA> \
-e DB_POSTGRESDB_PASSWORD=<POSTGRES_PASSWORD> \
-v n8n_data:/home/node/.n8n \
docker.n8n.io/n8nio/n8n
```

You can find a complete docker-compose file for PostgreSQL in the [n8n hosting repository](#).

Updating

To update n8n, in Docker Desktop, navigate to the **Images** tab and select **Pull** from the context menu to download the latest n8n image:

[Image: Docker Desktop]

You can also use the command line to pull the latest, or a specific version:

```
# Pull latest (stable) version
docker pull docker.n8n.io/n8nio/n8n

# Pull specific version
docker pull docker.n8n.io/n8nio/n8n:1.81.0

# Pull next (unstable) version
docker pull docker.n8n.io/n8nio/n8n:next
```

After pulling the updated image, stop your n8n container and start it again. You can also use the command line. Replace <container_id> in the commands below with the container ID you find in the first command:

```
# Find your container ID
docker ps -a

# Stop the container with the `<container_id>`
docker stop <container_id>

# Remove the container with the `<container_id>`
docker rm <container_id>

# Start the container
docker run --name=<container_name> [options] -d
docker.n8n.io/n8nio/n8n
```

Updating Docker Compose

-8<- “_snippets/self-hosting/installation/docker-compose-updating.md”
-8<- “_snippets/self-hosting/installation/tunnel.md”

Start n8n with --tunnel by running:

```
docker volume create n8n_data

docker run -it --rm \
--name n8n \
-p 5678:5678 \
-e GENERIC_TIMEZONE=<YOUR_TIMEZONE> \
-e TZ=<YOUR_TIMEZONE> \
-e N8N_ENFORCE_SETTINGS_FILE_PERMISSIONS=true \
-e N8N_RUNNERS_ENABLED=true \
-v n8n_data:/home/node/.n8n \
docker.n8n.io/n8nio/n8n \
start --tunnel
```

Next steps

- Find more information about Docker setup in the README file for the [Docker image](#). -8<- “_snippets/self-hosting/installation/server-setups-next-steps.md”
-

Server setups

Self-host with Docker Compose:

- [Digital Ocean](#)
- [Heroku](#)
- [Hetzner Cloud](#)

Self-host with Google Cloud Run (with access to n8n workflow tools for Google Workspace, e.g. Gmail, Drive):

- [Google Cloud Run](#)

Starting points for a Kubernetes setup:

- [AWS](#)
- [Azure](#)
- [Google Kubernetes Engine \(GKE\)](#)

Configuration guides to help you get started on other platforms:

- [Docker Compose](#)
-

Hosting n8n on DigitalOcean

This hosting guide shows you how to self-host n8n on a DigitalOcean droplet. It uses:

- [Caddy](#) (a reverse proxy) to allow access to the Droplet from the internet. Caddy will also automatically create and manage SSL / TLS certificates for your n8n instance.
- [Docker Compose](#) to create and define the application components and how they work together.

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

-8<- “_snippets/self-hosting/installation/latest-next-version.md”

Create a Droplet

1. [Log in](#) to DigitalOcean.
2. Select the project to host the Droplet, or [create a new project](#).
3. In your project, select **Droplets** from the **Manage** menu.
4. [Create a new Droplet](#) using the [Docker image](#) available on the **Marketplace** tab.

Log in to your Droplet and create new user

The rest of this guide requires you to log in to the Droplet using a terminal with SSH. Refer to [How to Connect to Droplets with SSH](#) for more information.

You should create a new user, to avoid working as the root user:

1. Log in as root.
2. Create a new user: `shell adduser <username>`
3. Follow the prompts in the CLI to finish creating the user.
4. Grant the new user administrative privileges: `shell usermod -aG sudo <username>` You can now run commands with superuser privileges by using `sudo` before the command.
5. Follow the steps to set up SSH for the new user: [Add Public Key Authentication](#).
6. Log out of the droplet.
7. Log in using SSH as the new user.

Clone configuration repository

Docker Compose, n8n, and Caddy require a series of folders and configuration files. You can clone these from [this repository](#) into the home folder of the logged-in user on your Droplet. The following steps will tell you which file to change and what changes to make.

Clone the repository with the following command:

```
git clone https://github.com/n8n-io/n8n-docker-caddy.git
```

And change directory to the root of the repository you cloned:

```
cd n8n-docker-caddy
```

Default folders and files

The host operating system (the DigitalOcean Droplet) copies the two folders you created to Docker containers to make them available to Docker. The two folders are:

- `caddy_config`: Holds the Caddy configuration files.
- `local_files`: A folder for files you upload or add using n8n.

Create Docker volumes

To persist the Caddy cache between restarts and speed up start times, create [a Docker volume](#) that Docker reuses between restarts:

```
sudo docker volume create caddy_data
```

Create a Docker volume for the n8n data:

```
sudo docker volume create n8n_data
```

Set up DNS

n8n typically operates on a subdomain. Create a DNS record with your provider for the subdomain and point it to the IP address of the Droplet. The exact steps for this depend on your DNS provider, but typically you need to create a new "A" record for the n8n subdomain. DigitalOcean provide [An Introduction to DNS Terminology, Components, and Concepts](#).

Open ports

n8n runs as a web application, so the Droplet needs to allow incoming access to traffic on port 80 for non-secure traffic, and port 443 for secure traffic.

Open the following ports in the Droplet's firewall by running the following two commands:

```
sudo ufw allow 80  
sudo ufw allow 443
```

Configure n8n

n8n needs some environment variables set to pass to the application running in the Docker container. The example `.env` file contains placeholders you need to replace with values of your own.

Open the file with the following command:

```
nano .env
```

The file contains inline comments to help you know what to change.

Refer to [Environment variables](#) for n8n environment variables details.

The Docker Compose file

The Docker Compose file (`docker-compose.yml`) defines the services the application needs, in this case Caddy and n8n.

- The Caddy service definition defines the ports it uses and the local volumes to copy to the containers.
- The n8n service definition defines the ports it uses, the environment variables n8n needs to run (some defined in the `.env` file), and the volumes it needs to copy to the containers.

The Docker Compose file uses the environment variables set in the `.env` file, so you shouldn't need to change its content, but to take a look, run the following command:

```
nano docker-compose.yml
```

Configure Caddy

Caddy needs to know which domains it should serve, and which port to expose to the outside world. Edit the `Caddyfile` file in the `caddy_config` folder.

```
nano caddy_config/Caddyfile
```

Change the placeholder domain to yours. If you followed the steps to name the subdomain n8n, your full domain is similar to `n8n.example.com`. The `n8n` in the `reverse_proxy` setting tells Caddy to use the service definition defined in the `docker-compose.yml` file:

```
n8n.<domain>.<suffix> {
    reverse_proxy n8n:5678 {
        flush_interval -1
    }
}
```

If you were to use `automate.example.com`, your `Caddyfile` may look something like:

```
automate.example.com {
    reverse_proxy n8n:5678 {
        flush_interval -1
    }
}
```

Start Docker Compose

Start n8n and Caddy with the following command:

```
sudo docker compose up -d
```

This may take a few minutes.

Test your setup

In your browser, open the URL formed of the subdomain and domain name defined earlier. Enter the user name and password defined earlier, and you should be able to access n8n.

Stop n8n and Caddy

You can stop n8n and Caddy with the following command:

```
sudo docker compose stop
```

Updating

-8<- “_snippets/self-hosting/installation/docker-compose-updating.md”

Next steps

-8<- “_snippets/self-hosting/installation/server-setups-next-steps.md”

Hosting n8n on Heroku

This hosting guide shows you how to self-host n8n on Heroku. It uses:

- [Docker Compose](#) to create and define the application components and how they work together.
- [Heroku’s PostgreSQL service](#) to host n8n’s data storage.
- A **Deploy to Heroku** button offering a one click, with minor configuration, deployment.

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

-8<- “_snippets/self-hosting/installation/latest-next-version.md”

Use the deployment template to create a Heroku project

The quickest way to get started with deploying n8n to Heroku is using the **Deploy to Heroku** button:

[\[Image: Deploy\]](#)

This opens the **Create New App** page on Heroku. Set a name for the project, and choose the region to deploy the project to.

Configure environment variables

Heroku pre-fills the configuration options defined in the `env` section of the `app.json` file, which also sets default values for the environment variables n8n uses.

You can change any of these values to suit your needs. You must change the following values:

- **N8N_ENCRYPTION_KEY**, which n8n uses to [encrypt user account details](#) before saving to the database.
- **WEBHOOK_URL** should match the application name you create to ensure that webhooks have the correct URL.

Deploy n8n

Select **Deploy app**.

After Heroku builds and deploys the app it provides links to **Manage App** or **View** the application.

Changing the deployment template

You can make changes to the deployment template by forking the [repository](#) and deploying from your fork.

The Dockerfile

By default the Dockerfile pulls the latest n8n image, if you want to use a different or fixed version, then update the image tag on the top line of the Dockerfile.

Heroku and exposing ports

Heroku doesn't allow Docker-based applications to define an exposed port with the `EXPOSE` command. Instead, Heroku provides a `PORT` environment variable that it dynamically populates at application runtime. The `entrypoint.sh` file overrides the default Docker image command to instead set the port variable that Heroku provides. You can then access n8n on port 80 in a web browser.

Configuring Heroku

The `heroku.yml` file defines the application you want to create on Heroku. It consists of two sections:

- `setup > addons` defines the Heroku addons to use. In this case, the PostgreSQL database addon.
- The `build` section defines how Heroku builds the application. In this case it uses the Docker buildpack to build a web service based on the supplied `Dockerfile`.

Next steps

-8<- “_snippets/self-hosting/installation/server-setups-next-steps.md”

Hosting n8n on Hetzner cloud

This hosting guide shows you how to self-host n8n on a Hetzner cloud server. It uses:

- Caddy (a reverse proxy) to allow access to the Server from the internet.
- Docker Compose to create and define the application components and how they work together.

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

-8<- “_snippets/self-hosting/installation/latest-next-version.md”

Create a server

1. Log in to the Hetzner Cloud Console.
2. Select the project to host the server, or create a new project by selecting **+ NEW PROJECT**.
3. Select **+ CREATE SERVER** on the project tile you want to add it to.

You can change most of the settings to suit your needs, but as this guide uses Docker to run the application, under the **Image** section, select “Docker CE” from the **APPS** tab.

Log in to your server

The rest of this guide requires you to log in to the server using a terminal with SSH. Refer to Access with SSH/rsync/BorgBackup for more information. You can find the public IP in the listing of the servers in your project.

Install Docker Compose

The Hetzner Docker app image doesn’t have Docker compose installed. Install it with the following commands:

```
apt update && apt -y upgrade  
apt install docker-compose-plugin
```

Clone configuration repository

Docker Compose, n8n, and Caddy require a series of folders and configuration files. You can clone these from this repository into the root user folder of the server. The following steps will tell you which file to change and what changes to make.

Clone the repository with the following command:

```
git clone https://github.com/n8n-io/n8n-docker-caddy.git
```

And change directory to the root of the repository you cloned:

```
cd n8n-docker-caddy
```

Default folders and files

The host operating system (the server) copies the two folders you created to Docker containers to make them available to Docker. The two folders are:

- `caddy_config`: Holds the Caddy configuration files.
- `local_files`: A folder for files you upload or add using n8n.

Create Docker volume

To persist the Caddy cache between restarts and speed up start times, create a Docker volume that Docker reuses between restarts:

```
docker volume create caddy_data
```

Create a Docker volume for the n8n data:

```
sudo docker volume create n8n_data
```

Set up DNS

n8n typically operates on a subdomain. Create a DNS record with your provider for the subdomain and point it to the IP address of the server. The exact steps for this depend on your DNS provider, but typically you need to create a new "A" record for the n8n subdomain. DigitalOcean provide An Introduction to DNS Terminology, Components, and Concepts.

Open ports

n8n runs as a web application, so the server needs to allow incoming access to traffic on port 80 for non-secure traffic, and port 443 for secure traffic.

Open the following ports in the server's firewall by running the following two commands:

```
sudo ufw allow 80  
sudo ufw allow 443
```

Configure n8n

n8n needs some environment variables set to pass to the application running in the Docker container. The example `.env` file contains placeholders you need to replace with values of your own.

Open the file with the following command:

```
nano .env
```

The file contains inline comments to help you know what to change.

Refer to [Environment variables](#) for n8n environment variables details.

The Docker Compose file

The Docker Compose file (`docker-compose.yml`) defines the services the application needs, in this case Caddy and n8n.

- The Caddy service definition defines the ports it uses and the local volumes to copy to the containers.
- The n8n service definition defines the ports it uses, the environment variables n8n needs to run (some defined in the `.env` file), and the volumes it needs to copy to the containers.

The Docker Compose file uses the environment variables set in the `.env` file, so you shouldn't need to change its content, but to take a look, run the following command:

```
nano docker-compose.yml
```

Configure Caddy

Caddy needs to know which domains it should serve, and which port to expose to the outside world. Edit the `Caddyfile` file in the `caddy_config` folder.

```
nano caddy_config/Caddyfile
```

Change the placeholder subdomain to yours. If you followed the steps to name the subdomain n8n, your full domain is similar to `n8n.example.com`. The `n8n` in the `reverse_proxy` setting tells Caddy to use the service definition defined in the `docker-compose.yml` file:

```
n8n.<domain>.<suffix> {
    reverse_proxy n8n:5678 {
        flush_interval -1
    }
}
```

Start Docker Compose

Start n8n and Caddy with the following command:

```
docker compose up -d
```

This may take a few minutes.

Test your setup

In your browser, open the URL formed of the subdomain and domain name defined earlier. Enter the user name and password defined earlier, and you should be able to access n8n.

Stop n8n and Caddy

You can stop n8n and Caddy with the following command:

```
sudo docker compose stop
```

Updating

```
-8<- "_snippets/self-hosting/installation/docker-compose-updating.md"
```

Next steps

```
-8<- "_snippets/self-hosting/installation/server-setups-next-steps.md"
```

Hosting n8n on Amazon Web Services

This hosting guide shows you how to self-host n8n with Amazon Web Services (AWS). It uses n8n with Postgres as a database backend using Kubernetes to manage the necessary resources and reverse proxy.

Hosting options

AWS offers several ways suitable for hosting n8n, including EC2 (virtual machines), and EKS (containers running with Kubernetes).

This guide uses [EKS](#) as the hosting option. Using Kubernetes requires some additional complexity and configuration, but is the best method for scaling n8n as demand changes.

Prerequisites

The steps in this guide use a mix of the AWS UI and [the eksctl CLI tool for EKS](#).

While not mentioned in the documentation for eksctl, you also need to [install the AWS CLI tool](#), and [configure authentication of the tool](#).

```
-8<- "_snippets/self-hosting/warning.md"
```

```
-8<- "_snippets/self-hosting/installation/latest-next-version.md"
```

Create a cluster

Use the eksctl tool to create a cluster specifying a name and a region with the following command:

```
eksctl create cluster --name n8n --region <your-aws-region>
```

This can take a while to create the cluster.

Once the cluster is created, eksctl automatically sets the kubectl context to the cluster.

Clone configuration repository

Kubernetes and n8n require a series of configuration files. You can clone these from [this repository](#). The following steps tell you what each file does, and what settings you need to change.

Clone the repository with the following command:

```
git clone https://github.com/n8n-io/n8n-hosting.git
```

And change directory:

```
cd n8n-hosting/kubernetes
```

Configure Postgres

For larger scale n8n deployments, Postgres provides a more robust database backend than SQLite.

Configure volume for persistent storage

To maintain data between pod restarts, the Postgres deployment needs a persistent volume. The default AWS storage class, `gp3`, is suitable for this purpose. This is defined in the `postgres-claim0-persistentvolumeclaim.yaml` manifest.

```
...
spec:
  storageClassName: gp3
  accessModes:
    - ReadWriteOnce
...

```

Postgres environment variables

Postgres needs some environment variables set to pass to the application running in the containers.

The example `postgres-secret.yaml` file contains placeholders you need to replace with values of your own for user details and the database to use.

PostgreSQL uses a root user (`POSTGRES_USER`) for setup and administration, but it's best practice to create a separate non-root user (`POSTGRES_NON_ROOT_USER`) for n8n. The root user has full control, while n8n only needs the non-root user permissions to run. Configuring both improves security and helps prevent accidental changes to the database system.

The `postgres-deployment.yaml` manifest then uses the values from this manifest file to send to the application pods.

Configure n8n

Create a volume for file storage

While not essential for running n8n, using persistent volumes helps maintain files uploaded while using n8n and if you want to persist [manual n8n encryption keys](#) between restarts, which saves a file containing the key into file storage during startup.

The `n8n-claim0-persistentvolumeclaim.yaml` manifest creates this, and the n8n Deployment mounts that claim in the `volumes` section of the `n8n-deployment.yaml` manifest.

```
...
volumes:
  - name: n8n-claim0
    persistentVolumeClaim:
      claimName: n8n-claim0
...

```

Pod resources

[Kubernetes](#) lets you specify the minimum resources application containers need and the limits they can run to. The example YAML files cloned above contain the following in the `resources` section of the `n8n-deployment.yaml` file:

```
...
resources:
  requests:
    memory: "250Mi"
  limits:
    memory: "500Mi"
...

```

This defines a minimum of 250mb per container, a maximum of 500mb, and lets Kubernetes handle CPU. You can change these values to match your own needs. As a guide, here are the resources values for the n8n cloud offerings:

```
-8<- " snippets/self-hosting/installation/suggested-pod-resources.md"
```

Optional: Environment variables

You can configure n8n settings and behaviors using environment variables.

Create an `n8n-secret.yaml` file. Refer to [Environment variables](#) for n8n environment variables details.

Deployments

The two deployment manifests (`n8n-deployment.yaml` and `postgres-deployment.yaml`) define the n8n and Postgres applications to Kubernetes.

The manifests define the following:

- Send the environment variables defined to each application pod
- Define the container image to use
- Set resource consumption limits
- The volumes defined earlier and `volumeMounts` to define the path in the container to mount volumes.
- Scaling and restart policies. The example manifests define one instance of each pod. You should change this to meet your needs.

Services

The two service manifests (`postgres-service.yaml` and `n8n-service.yaml`) expose the services to the outside world using the Kubernetes load balancer using ports 5432 and 5678 respectively by default.

Send to Kubernetes cluster

Send all the manifests to the cluster by running the following command in the `n8n-kubernetes-hosting` directory:

```
kubectl apply -f .
```

Set up DNS

`n8n` typically operates on a subdomain. Create a DNS record with your provider for the subdomain and point it to a static address of the instance.

To find the address of the `n8n` service running on the instance:

1. Open the **Clusters** section of the **Amazon Elastic Kubernetes Service** page in the AWS console.
2. Select the name of the cluster to open its configuration page.
3. Select the **Resources** tab, then **Service and networking > Services**.
4. Select the **n8n** service and copy the **Load balancer URLs** value. Use this value suffixed with the `n8n` service port (5678) for DNS.

Delete resources

If you need to delete the setup, you can remove the resources created by the manifests with the following command:

```
kubectl delete -f .
```

Next steps

-8<- “_snippets/self-hosting/installation/server-setups-next-steps.md”

Hosting n8n on Azure

This hosting guide shows you how to self-host n8n on Azure. It uses n8n with Postgres as a database backend using Kubernetes to manage the necessary resources and reverse proxy.

Prerequisites

You need [The Azure command line tool](#)

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

-8<- “_snippets/self-hosting/installation/latest-next-version.md”

Hosting options

Azure offers several ways suitable for hosting n8n, including Azure Container Instances (optimized for running containers), Linux Virtual Machines, and Azure Kubernetes Service (containers running with Kubernetes).

This guide uses the Azure Kubernetes Service (AKS) as the hosting option. Using Kubernetes requires some additional complexity and configuration, but is the best method for scaling n8n as demand changes.

The steps in this guide use a mix of the Azure UI and command line tool, but you can use either to accomplish most tasks.

Open the Azure Kubernetes Service

From [the Azure portal](#) select **Kubernetes services**.

Create a cluster

From the Kubernetes services page, select **Create > Create a Kubernetes cluster**.

You can select any of the configuration options that suit your needs, then select **Create** when done.

Set Kubectl context

The remainder of the steps in this guide require you to set the Azure instance as the Kubectl context. You can find the connection details for a cluster instance by opening its details page and then the **Connect** button. The resulting code snippets shows the steps to paste and run into a terminal to change your local Kubernetes settings to use the new cluster.

Clone configuration repository

Kubernetes and n8n require a series of configuration files. You can clone these from [this repository](#). The following steps tell you which file configures what and what you need to change.

Clone the repository with the following command:

```
git clone https://github.com/n8n-io/n8n-hosting.git
```

And change directory:

```
cd n8n-hosting/kubernetes
```

Configure Postgres

For larger scale n8n deployments, Postgres provides a more robust database backend than SQLite.

Configure volume for persistent storage

To maintain data between pod restarts, the Postgres deployment needs a persistent volume. The default storage class is suitable for this purpose and is defined in the `postgres-claim0-persistentvolumeclaim.yaml` manifest.

Postgres environment variables

Postgres needs some environment variables set to pass to the application running in the containers.

The example `postgres-secret.yaml` file contains placeholders you need to replace with your own values. Postgres will use these details when creating the database..

The `postgres-deployment.yaml` manifest then uses the values from this manifest file to send to the application pods.

Configure n8n

Create a volume for file storage

While not essential for running n8n, using persistent volumes is required for:

- Using nodes that interact with files, such as the binary data node.
- If you want to persist manual n8n encryption keys between restarts. This saves a file containing the key into file storage during startup.

The `n8n-claim0-persistentvolumeclaim.yaml` manifest creates this, and the n8n Deployment mounts that claim in the `volumes` section of the `n8n-deployment.yaml` manifest.

```
...
volumes:
  - name: n8n-claim0
    persistentVolumeClaim:
      claimName: n8n-claim0
...

```

Pod resources

[Kubernetes lets you](#) optionally specify the minimum resources application containers need and the limits they can run to. The example YAML files cloned above contain the following in the resources section of the n8n-deployment.yaml file:

```
...  
  resources:  
    requests:  
      memory: "250Mi"  
    limits:  
      memory: "500Mi"  
...
```

This defines a minimum of 250mb per container, a maximum of 500mb, and lets Kubernetes handle CPU. You can change these values to match your own needs. As a guide, here are the resources values for the n8n cloud offerings:

```
-8<- "_snippets/self-hosting/installation/suggested-pod-resources.md"
```

Optional: Environment variables

You can configure n8n settings and behaviors using environment variables.

Create an n8n-secret.yaml file. Refer to [Environment variables](#) for n8n environment variables details.

Deployments

The two deployment manifests (n8n-deployment.yaml and postgres-deployment.yaml) define the n8n and Postgres applications to Kubernetes.

The manifests define the following:

- Send the environment variables defined to each application pod
- Define the container image to use
- Set resource consumption limits with the resources object
- The volumes defined earlier and volumeMounts to define the path in the container to mount volumes.
- Scaling and restart policies. The example manifests define one instance of each pod. You should change this to meet your needs.

Services

The two service manifests (postgres-service.yaml and n8n-service.yaml) expose the services to the outside world using the Kubernetes load balancer using ports 5432 and 5678 respectively.

Send to Kubernetes cluster

Send all the manifests to the cluster with the following command:

```
kubectl apply -f .
```

Set up DNS

n8n typically operates on a subdomain. Create a DNS record with your provider for the subdomain and point it to the IP address of the n8n service. Find the IP address of the n8n service from the **Services & ingresses** menu item of the cluster you want to use under the **External IP** column. You need to add the n8n port, “5678” to the URL.

Delete resources

Remove the resources created by the manifests with the following command:

```
kubectl delete -f .
```

Next steps

```
-8<- "_snippets/self-hosting/installation/server-setups-next-steps.md"
```

Hosting n8n on Google Cloud Run

This hosting guide shows you how to self-host n8n on Google Cloud Run, a serverless container runtime. If you’re just getting started with n8n and don’t need a production-grade deployment, you can go with the “easy mode” option below for deployment. Otherwise, if you intend to use this n8n deployment at-scale, refer to the “durable mode” instructions further down.

You can also enable access via OAuth to Google Workspace, such as Gmail and Drive, to use these services as n8n workflow tools. Instructions for granting n8n access to these services are at the end of this documentation.

If you want to deploy to Google Kubernetes Engine (GKE) instead, you can refer to [these instructions](#).

```
-8<- "_snippets/self-hosting/warning.md"
```

```
-8<- "_snippets/self-hosting/installation/latest-next-version.md"
```

Before you begin: get a Google Cloud project

If you have not yet created a Google Cloud project, [do this first](#) (and ensure you have billing enabled on the project; even if your Cloud Run service runs for free you must have billing activated to deploy). Otherwise, navigate to the project where you want to deploy n8n.

Easy mode

This is the fastest way to deploy n8n on Cloud Run. For this deployment, n8n's data is in-memory so this is only recommended for demo purposes. **Anytime this Cloud Run service scales to zero or is redeployed, the n8n data will be lost.** Refer to the durable mode instructions below if you need a production-grade deployment.

If you have not yet created a Google Cloud project, [do this first](#) (and ensure you have billing enabled on the project; even if your Cloud Run service will run for free you must have billing enabled to activate it to deploy). Otherwise, navigate to the project where you want to deploy n8n.

Open the Cloud Shell Terminal (on the Google Cloud console, either type "G" then "S" or click on the terminal icon on the upper right).

Once your session is open, you may need to run this command first to login (and follow the steps it asks you to complete):

```
gcloud auth login
```

You can also explicitly enable the Cloud Run API (even if you don't do this, it will ask if you want this enabled when you deploy):

```
gcloud services enable run.googleapis.com
```

To deploy n8n:

```
gcloud run deploy n8n \
  --image=n8nio/n8n \
  --region=us-west1 \
  --allow-unauthenticated \
  --port=5678 \
  --no-cpu-throttling \
  --memory=2Gi
```

(you can specify whichever region you prefer, instead of "us-west1")

Once the deployment finishes, open another tab to navigate to the Service URL. n8n may still be loading and you will see a "n8n is starting up. Please wait" message, but shortly thereafter you should see the n8n login screen.

Optional: If you want to keep this n8n service running for as long as possible to avoid data loss, you can also set manual scale to 1 to prevent it from autoscaling to 0.

```
gcloud run deploy n8n \
  --image=n8nio/n8n \
  --region=us-west1 \
  --allow-unauthenticated \
  --port=5678 \
  --no-cpu-throttling \
  --memory=2Gi \
  --scaling=1
```

This does not prevent data loss completely, such as whenever the Cloud Run service is re-deployed/updated. If you want truly persistent data, you should refer to the instructions below for how to attach a database.

Durable mode

The following instructions are intended for a more durable, production-grade deployment of n8n on Cloud Run. It includes resources such as a database for persistence and secret manager for sensitive data.

If you want to deploy the following setup via Terraform, refer to this [example](#) which deploys the same setup as the following (without the OAuth setup for Google Workspace tools).

Enable APIs and set env vars

Open the Cloud Shell Terminal (on the Google Cloud console, either type “G” then “S” or click on the terminal icon on the upper right) and run these commands in the terminal session:

```
## You may need to login first
gcloud auth login

gcloud services enable run.googleapis.com
gcloud services enable sqladmin.googleapis.com
gcloud services enable secretmanager.googleapis.com
```

You’ll also want to set some environment variables for the remainder of these instructions:

```
export PROJECT_ID=your-project
export REGION=region-where-you-want-this-deployed
```

Setup your Postgres database

Run this command to create the Postgres DB instance (it will take a few minutes to complete; also ensure you update the root-password field with your own desired password):

```
gcloud sql instances create n8n-db \
    --database-version=POSTGRES_13 \
    --tier=db-f1-micro \
    --region=$REGION \
    --root-password="change-this-password" \
    --storage-size=10GB \
    --availability-type=ZONAL \
    --no-backup \
    --storage-type=HDD
```

Once complete, you can add the database that n8n will use:

```
gcloud sql databases create n8n --instance=n8n-db
```

Create the DB user for n8n (change the password value, of course):

```
gcloud sql users create n8n-user \
    --instance=n8n-db \
    --password="change-this-password"
```

You can save the password you set for this n8n-user to a file for the next step of saving the password in Secret Manager. Be sure to delete this file later.

Store sensitive data in Secret Manager

While not required, it's absolutely recommended to store your sensitive data in Secrets Manager.

Create a secret for the database password (replace "/your/password/file" with the file you created above for the n8n-user password):

```
gcloud secrets create n8n-db-password \
    --data-file=/your/password/file \
    --replication-policy="automatic"
```

Create an encryption key (you can use your own, this example generates a random one):

```
openssl rand -base64 -out my-encryption-key 42
```

Create a secret for this encryption key (replace "my-encryption-key" if you are supplying your own):

```
gcloud secrets create n8n-encryption-key \
    --data-file=my-encryption-key \
    --replication-policy="automatic"
```

Now you can delete my-encryption-key and the database password files you created. These values are now securely stored in Secret Manager.

Create a service account for Cloud Run

You want this Cloud Run service to be restricted to access only the resources it needs. The following commands create the service account and adds the permissions necessary to access secrets and the database:

```
gcloud iam service-accounts create n8n-service-account \
    --display-name="n8n Service Account"

gcloud secrets add-iam-policy-binding n8n-db-password \
    --member="serviceAccount:n8n-service-
account@$PROJECT_ID.iam.gserviceaccount.com" \
    --role="roles/secretmanager.secretAccessor"

gcloud secrets add-iam-policy-binding n8n-encryption-key \
    --member="serviceAccount:n8n-service-
account@$PROJECT_ID.iam.gserviceaccount.com" \
    --role="roles/secretmanager.secretAccessor"

gcloud projects add-iam-policy-binding $PROJECT_ID \
    --member="serviceAccount:n8n-service-
account@$PROJECT_ID.iam.gserviceaccount.com" \
    --role="roles/cloudsql.client"
```

Deploy the Cloud Run service

Now you can deploy your n8n service:

```
gcloud run deploy n8n \
```

```

--image=n8nio/n8n:latest \
--command="/bin/sh" \
--args="-c,sleep 5;n8n start" \
--region=$REGION \
--allow-unauthenticated \
--port=5678 \
--memory=2Gi \
--no-cpu-throttling \
--set-env-
vars="N8N_PORT=5678,N8N_PROTOCOL=https,DB_TYPE=postgresdb,DB_POSTGRE
SDB_DATABASE=n8n,DB_POSTGRESDB_USER=n8n-
user,DB_POSTGRESDB_HOST=/cloudsql/$PROJECT_ID:$REGION:n8n-
db,DB_POSTGRESDB_PORT=5432,DB_POSTGRESDB_SCHEMA=public,GENERIC_TIMEZ
ONE=UTC,QUEUE_HEALTH_CHECK_ACTIVE=true" \
--set-secrets="DB_POSTGRESDB_PASSWORD=n8n-db-
password:latest,N8N_ENCRYPTION_KEY=n8n-encryption-key:latest" \
--add-cloudsql-instances=$PROJECT_ID:$REGION:n8n-db \
--service-account=n8n-service-
account@$PROJECT_ID.iam.gserviceaccount.com

```

Once the deployment finishes, open another tab to navigate to the Service URL. You should see the n8n login screen.

Troubleshooting

If you see a “Cannot GET /” screen this usually indicates that n8n is still starting up. You can refresh the page and it should eventually load.

(Optional) Enabling Google Workspace services as n8n tools

If you want to use Google Workspace services (Gmail, Calendar, Drive, etc.) as tools in n8n, it’s recommended to setup OAuth to access these services.

First ensure the respective APIs you want are enabled:

```

## Enable whichever APIs you need
## Note: If you want Sheets/Docs, it's not enough to just enable
Drive; these services each have their own API
gcloud services enable gmail.googleapis.com
gcloud services enable drive.googleapis.com
gcloud services enable sheets.googleapis.com
gcloud services enable docs.googleapis.com
gcloud services enable calendar-json.googleapis.com

```

Re-deploy n8n on Cloud Run with the necessary OAuth callback URLs as environment variables:

```

export SERVICE_URL="your-n8n-service-URL"
## e.g. https://n8n-12345678.us-west1.run.app

gcloud run services update n8n \
--region=$REGION \
--update-env-vars="N8N_HOST=$(echo $SERVICE_URL | sed
's/https:\//'),WEBHOOK_URL=$SERVICE_URL,N8N_EDITOR_BASE_URL=$SERV
ICE_URL"

```

Lastly, you must setup OAuth for these services. Visit <https://console.cloud.google.com/auth> and follow these steps:

1. Click “Get Started” if this button shows (when you have not yet setup OAuth in this Cloud project).
 2. For “App Information”, enter whichever “App Name” and “User Support Email” you prefer.
 3. For “Audience”, select “Internal” if you intend to only enable access to your user(s) within this same Google Workspace. Otherwise, you can select “External”.
 4. Enter “Contact Information”.
 5. If you selected “External”, then click “Audience” and add any test users you need to grant access.
 6. Click “Clients” > “Create client”, select “Web application” for “Application type”, enter your n8n service URL into “Authorized JavaScript origins”, and “/rest/oauth2-credential/callback” into “Authorized redirect URIs” where your YOUR-N8N-URL is also the n8n service URL (e.g. <https://n8n-12345678.us-west1.run.app/rest/oauth2-credential/callback>). Make sure you download the created client’s JSON file since it contains the client secret which you will not be able to see later in the Console.
 7. Click “Data Access” and add the scopes you want n8n to have access for (e.g. to access Google Sheets, you need <https://googleapis.com/auth/drive.file> and <https://googleapis.com/auth/spreadsheets>)
 8. Now you should be able to use these workspace services. You can test if it works by logging into n8n, add a Tool for the respective service and add its credentials using the information in the OAuth client JSON file from step 6.
-

Hosting n8n on Google Kubernetes Engine

Google Cloud offers several options suitable for hosting n8n, including Cloud Run (optimized for running containers), Compute Engine (VMs), and Kubernetes Engine (containers running with Kubernetes).

This guide uses the Google Kubernetes Engine (GKE) as the hosting option. If you want to use Cloud Run, refer to [these instructions](#).

Most of the steps in this guide use the Google Cloud UI, but you can also use the [gcloud command line tool](#) instead to undertake all the steps.

Prerequisites

- The [gcloud command line tool](#)
 - The [gke-gcloud-auth-plugin](#) (install the gcloud CLI first)
- 8<- “_snippets/self-hosting/warning.md”
- 8<- “_snippets/self-hosting/installation/latest-next-version.md”

Create project

GCP encourages you to create projects to logically organize resources and configuration. Create a new project for your n8n deployment from your Google Cloud Console: select the project dropdown menu and then the **NEW PROJECT** button. Then select the newly created project. As you follow the other steps in this guide, make sure you have the correct project selected.

Enable the Kubernetes Engine API

GKE isn't enabled by default. Search for "Kubernetes" in the top search bar and select "Kubernetes Engine" from the results.

Select **ENABLE** to enable the Kubernetes Engine API for this project.

Create a cluster

From the [GKE service page](#), select **Clusters > CREATE**. Make sure you select the "Standard" cluster option, n8n doesn't work with an "Autopilot" cluster. You can leave the cluster configuration on defaults unless there's anything specifically you need to change, such as location.

Set Kubectl context

The rest of the steps in this guide require you to set the GCP instance as the Kubectl context. You can find the connection details for a cluster instance by opening its details page and selecting **CONNECT**. The displayed code snippet shows a connection string for the gcloud CLI tool. Paste and run the code snippet in the gcloud CLI to change your local Kubernetes settings to use the new gcloud cluster.

Clone configuration repository

Kubernetes and n8n require a series of configuration files. You can clone these from [this repository](#) locally. The following steps explain the file configuration and how to add your information.

Clone the repository with the following command:

```
git clone https://github.com/n8n-io/n8n-hosting.git
```

And change directory:

```
cd n8n-hosting/kubernetes
```

Configure Postgres

For larger scale n8n deployments, Postgres provides a more robust database backend than SQLite.

Create a volume for persistent storage

To maintain data between pod restarts, the Postgres deployment needs a persistent volume. Running Postgres on GCP requires a specific Kubernetes Storage Class. You can read [this guide](#) for specifics, but the `storage.yaml` manifest creates it for you. You may want to change the regions to create the storage in under the `allowedTopologies > matchLabelExpressions > values` key. By default, they're set to `us-central`.

```
...
  allowedTopologies:
    - matchLabelExpressions:
      - key: failure-domain.beta.kubernetes.io/zone
        values:
          - us-central1-b
          - us-central1-c
```

Postgres environment variables

Postgres needs some environment variables set to pass to the application running in the containers.

The example `postgres-secret.yaml` file contains placeholders you need to replace with your own values. Postgres will use these details when creating the database..

The `postgres-deployment.yaml` manifest then uses the values from this manifest file to send to the application pods.

Configure n8n

Create a volume for file storage

While not essential for running n8n, using persistent volumes is required for:

- Using nodes that interact with files, such as the binary data node.
- If you want to persist [manual n8n encryption keys](#) between restarts. This saves a file containing the key into file storage during startup.

The `n8n-claim0-persistentvolumeclaim.yaml` manifest creates this, and the n8n Deployment mounts that claim in the `volumes` section of the `n8n-deployment.yaml` manifest.

```
...
  volumes:
    - name: n8n-claim0
      persistentVolumeClaim:
        claimName: n8n-claim0
...

```

Pod resources

[Kubernetes lets you](#) optionally specify the minimum resources application containers need and the limits they can run to. The example YAML files cloned above contain the following in the `resources` section of the `n8n-deployment.yaml` and `postgres-deployment.yaml` files:

```
...
resources:
  requests:
    memory: "250Mi"
  limits:
    memory: "500Mi"
...
```

This defines a minimum of 250mb per container, a maximum of 500mb, and lets Kubernetes handle CPU. You can change these values to match your own needs. As a guide, here are the resources values for the n8n cloud offerings:

```
-8<- "_snippets/self-hosting/installation/suggested-pod-resources.md"
```

Optional: Environment variables

You can configure n8n settings and behaviors using environment variables.

Create an `n8n-secret.yaml` file. Refer to [Environment variables](#) for n8n environment variables details.

Deployments

The two deployment manifests (`n8n-deployment.yaml` and `postgres-deployment.yaml`) define the n8n and Postgres applications to Kubernetes.

The manifests define the following:

- Send the environment variables defined to each application pod
- Define the container image to use
- Set resource consumption limits with the `resources` object
- The `volumes` defined earlier and `volumeMounts` to define the path in the container to mount volumes.
- Scaling and restart policies. The example manifests define one instance of each pod. You should change this to meet your needs.

Services

The two service manifests (`postgres-service.yaml` and `n8n-service.yaml`) expose the services to the outside world using the Kubernetes load balancer using ports 5432 and 5678 respectively.

Send to Kubernetes cluster

Send all the manifests to the cluster with the following command:

```
kubectl apply -f .
```

Set up DNS

n8n typically operates on a subdomain. Create a DNS record with your provider for the subdomain and point it to the IP address of the n8n service. Find the IP address of the n8n service from the **Services & Ingress** menu item of the cluster you want to use under the **Endpoints** column.

Delete resources

Remove the resources created by the manifests with the following command:

```
kubectl delete -f .
```

Next steps

-8<- “_snippets/self-hosting/installation/server-setups-next-steps.md”

Docker-Compose

These instructions cover how to run n8n on a Linux server using Docker Compose.

If you have already installed Docker and Docker-Compose, then you can start with [step 3](#).

You can find Docker Compose configurations for various architectures in the [n8n-hosting repository](#).

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

-8<- “_snippets/self-hosting/installation/latest-next-version.md”

1. Install Docker and Docker Compose

The way that you install Docker and Docker Compose depends on your Linux distribution. You can find specific instructions for each component in the links below:

- [Docker Engine](#)
- [Docker Compose](#)

After following the installation instructions, verify that Docker and Docker Compose are available by typing:

```
docker --version  
docker compose version
```

2. Optional: Non-root user access

You can optionally grant access to run Docker without the `sudo` command.

To grant access to the user that you're currently logged in with (assuming they have sudo access), run:

```
sudo usermod -aG docker ${USER}
# Register the `docker` group membership with current session
without changing your primary group
exec sg docker newgrp
```

To grant access to a different user, type the following, substituting <USER_TO_RUN_DOCKER> with the appropriate username:

```
sudo usermod -aG docker <USER_TO_RUN_DOCKER>
```

You will need to run exec sg docker newgrp from any of that user's existing sessions for it to access the new group permissions.

You can verify that your current session recognizes the docker group by typing:

```
groups
```

3. DNS setup

To host n8n online or on a network, create a dedicated subdomain pointed at your server.

Add an A record to route the subdomain accordingly:

| Record type | Name | Destination |
|-------------|---------------------------------|--------------------------|
| A | n8n (or your desired subdomain) | <your_server_IP_address> |

4. Create an .env file

Create a project directory to store your n8n environment configuration and Docker Compose files and navigate inside:

```
mkdir n8n-compose
cd n8n-compose
```

Inside the n8n-compose directory, create an .env file to customize your n8n instance's details. Change it to match your own information:

```
```shell title=".env file" # DOMAIN_NAME and SUBDOMAIN
together determine where n8n will be reachable from # The top level
domain to serve from DOMAIN_NAME=example.com
```

## The subdomain to serve from

```
SUBDOMAIN=n8n
```

**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**

GENERIC\_TIMEZONE=Europe/Berlin

## **The email address to use for the TLS/SSL certificate creation**

SSL\_EMAIL=user@example.com

## 5. Create local files directory

Inside your project directory, create a directory called `local-files` for sharing files between the n8n instance and the host system (for example, using the [Read/Write Files from Disk node] (/integrations/builtin/core-nodes/n8n-nodes-base.readwritefile.md)):

```
```shell
mkdir local-files
```

The Docker Compose file below can automatically create this directory, but doing it manually ensures that it's created with the right ownership and permissions.

6. Create Docker Compose file

Create a `compose.yaml` file. Paste the following in the file:

```
```yaml title="compose.yaml file" services: traefik: image: "traefik"
restart: always command: - "-api.insecure=true" - "-"
providers.docker=true" - "-providers.docker.exposedbydefault=false"
- "-entrypoints.web.address=:80" - "-"
entrypoints.web.http.redirects.entryPoint.to=websecure" - "-"
entrypoints.web.http.redirects.entrypoint.scheme=https" - "-"
entrypoints.websecure.address=:443" - "-"
certificatesresolvers.mytlschallenge.acme.tlschallenge=true" - "-"
certificatesresolvers.mytlschallenge.acme.email=${SSL_EMAIL}" - "-"
certificatesresolvers.mytlschallenge.acme.storage=/letsencrypt/acme.json"
ports: - "80:80" - "443:443" volumes: - traefik_data:/letsencrypt
/var/run/docker.sock:/var/run/docker.sock:ro

n8n: image: docker.n8n.io/n8nio/n8n restart: always ports: -
"127.0.0.1:5678:5678" labels: - traefik.enable=true -
traefik.http.routers.n8n.rule=Host(${SUBDOMAIN}.${DOMAIN_NAME}) -
traefik.http.routers.n8n.tls=true -
traefik.http.routers.n8n.entrypoints=web,websecure -
```

```

traefik.http.routers.n8n.tls.certresolver=mytlschallenge -
traefik.http.middlewares.n8n.headers.SSLRedirect=true -
traefik.http.middlewares.n8n.headers.STSSeconds=315360000 -
traefik.http.middlewares.n8n.headers.browserXSSFilter=true -
traefik.http.middlewares.n8n.headers.contentTypeNosniff=true -
traefik.http.middlewares.n8n.headers.forceSTSHeader=true -
traefik.http.middlewares.n8n.headers.SSLHost=DOMAINNAME
- traefik.http.middlewares.n8n.headers.STSIncludeSubdomains
= true - traefik.http.middlewares.n8n.headers.STSPreload = true
- traefik.http.routers.n8n.middlewares = n8n@dockerenvironment :
- N8NFORCESETTINGSFILEPERMISSIONS = true - N8NHOST=
{SUBDOMAIN}.DOMAINNAME - N8NPORT = 5678
- N8NPROTOCOL = https - N8NRUNNERSENABLED = true
- NODEENV = production - WEBHOOKURL
= https://{{SUBDOMAIN}}.DOMAINNAME/ - GENERICTIMEZONE=
{GENERIC_TIMEZONE} - TZ=${GENERIC_TIMEZONE} volumes: -
n8n_data:/home/node/.n8n - ./local-files:/files

volumes: n8n_data: traefik_data:

```

The Docker Compose file above configures two containers: one for n8n, and one to run [traefik](<https://github.com/traefik/traefik>), an application proxy to manage TLS/SSL certificates and handle routing.

It also creates and mounts two [Docker Volumes] (<https://docs.docker.com/engine/storage/volumes/>) and mounts the `local-files` directory you created earlier:

Name	Type
Container mount	Description
`n8n_data`	[Volume] ( <a href="https://docs.docker.com/engine/storage/volumes/">https://docs.docker.com/engine/storage/volumes/</a> )   `/home/node/.n8n`   Where n8n saves its SQLite database file and encryption key.
`traefik_data`	[Volume] ( <a href="https://docs.docker.com/engine/storage/volumes/">https://docs.docker.com/engine/storage/volumes/</a> )   `/letsencrypt`   Where traefik saves TLS/SSL certificate data.
`.local-files`	[Bind] ( <a href="https://docs.docker.com/engine/storage/bind-mounts/">https://docs.docker.com/engine/storage/bind-mounts/</a> )   `/files`   A local directory shared between the n8n instance and host. In n8n, use the `/files` path to read from and write to this directory.

## 7. Start Docker Compose

Start n8n by typing:

```

```shell
sudo docker compose up -d

```

To stop the containers, type:

```
sudo docker compose stop
```

8. Done

You can now reach n8n using the subdomain + domain combination you defined in your .env file configuration. The above example would result in <https://n8n.example.com>.

n8n is only accessible using secure HTTPS, not over plain HTTP.

If you have trouble reaching your instance, check your server's firewall settings and your DNS configuration.

Next steps

-8<- “_snippets/self-hosting/installation/server-setups-next-steps.md”

Update self-hosted n8n

It's important to keep your n8n version up to date. This ensures you get the latest features and fixes.

Some tips when updating:

-8<- “_snippets/manage-cloud/updating-best-practices.md”

For instructions on how to update, refer to the documentation for your installation method:

- [Installed with npm](#)
 - [Installed with Docker](#)
-

Environment variables overview

This section lists the environment variables that you can use to change n8n's configuration settings when self-hosting n8n.

For the complete and most up-to-date list of all environment variables, see the full reference in the n8n docs repository: [Full environment variables index](#).

AI Assistant environment variables

Variable	Type	Default	Description
N8N_AI_ASSISTANT_BASE_URL	String	(empty)	Base URL of the AI assistant service, specified as <code>https://ai-assistant.n8n.io</code> .

Required if you self-host n8n and want to enable the AI Assistant.

Binary data environment variables

-8<- “ snippets/self-hosting/file-based-configuration.md”

By default, n8n uses memory to store binary data. Enterprise users can choose to use an external service instead. Refer to [External storage](#) for more information on using external storage for binary data.

Variable	Type	Default
N8N_AVAILABLE_BINARY_DATA_MODES	String	filesystem
N8N_BINARY_DATA_STORAGE_PATH	String	N8N_USER_FOLDER/binaryData
N8N_DEFAULT_BINARY_DATA_MODE	String	default

Credentials environment variables

-8<- “_snippets/self-hosting/file-based-configuration.md”

Enable credential overwrites using the following environment variables. Refer to [Credential overwrites](#) for details.

Variable	Type	Default	Description
CREDENTIALS_OVERWRITE_DATA/_FILE	*	-	Overwrites credential file.
CREDENTIALS_OVERWRITE_ENDPOINT	String	-	The API endpoint to fetch credentials.
CREDENTIALS_OVERWRITE_PERSISTENCE	Boolean	false	Enable database persistence credential overwrites. Required for multiinstance queue mode propagate overwrites workers through a publish/subscribe approach.
CREDENTIALS_DEFAULT_NAME	String	My credentials	The default name for credentials.

Database environment variables

-8<- “_snippets/self-hosting/file-based-configuration.md”

By default, n8n uses SQLite. n8n also supports PostgreSQL. n8n [deprecated support for MySQL and MariaDB](#) in v1.0.

This page outlines environment variables to configure your chosen database for your self-hosted n8n instance.

Variable	Type	Default	Description
DB_TYPE/_FILE	Enum string: sqlite	sqlite	The database to use.

		postgresdb	
DB_TABLE_PREFIX	*	-	Prefix to use for table names.
DB_PING_INTERVAL_SECONDS	Number	2	The interval, in seconds, between pings to the database to check if the connection is still alive.

PostgreSQL

Variable	Type	Default	Description
DB_POSTGRESDB_DATABASE/_FILE	String	n8n	The name of the PostgreSQL database.
DB_POSTGRESDB_HOST/_FILE	String	localhost	The PostgreSQL host address.
DB_POSTGRESDB_PORT/_FILE	Number	5432	The PostgreSQL port number.
DB_POSTGRESDB_USER/_FILE	String	postgres	The PostgreSQL user name.
DB_POSTGRESDB_PASSWORD/_FILE	String	-	The PostgreSQL password.
DB_POSTGRESDB_POOL_SIZE/_FILE	Number	2	Control parallel connections. Should be increased with resource utilization, many connections will degrade performance.
DB_POSTGRESDB_CONNECTION_TIMEOUT/_FILE	Number	20000	PostgreSQL connection timeout.
DB_POSTGRESDB_IDLE_CONNECTION_TIMEOUT/_FILE	Number	30000	Amount of time an idle connection is eligible to be closed for being reused.
DB_POSTGRESDB_SCHEMA/_FILE	String	public	The PostgreSQL schema.
DB_POSTGRESDB_SSL_ENABLED/_FILE	Boolean	false	Whether SSL is enabled or not. DB_POSTGRESDB_SSL_CA, DB_POSTGRESDB_SSL_CERT and DB_POSTGRESDB_SSL_KEY are defined.
DB_POSTGRESDB_SSL_CA/_FILE	String	-	The PostgreSQL SSL certificate file.
DB_POSTGRESDB_SSL_CERT/_FILE	String	-	The PostgreSQL SSL certificate file.
DB_POSTGRESDB_SSL_KEY/_FILE	String	-	The PostgreSQL SSL key file.

			Summary
/_FILE			key. If n8n s1 unautho connect not (fals
DB_POSTGRESDB_SSL_REJECT_UNAUTHORIZED	Boolean	true	
/_FILE			

SQLite

Variable	Type	Default	Description
DB_SQLITE_POOL_SIZE	Number	0	Controls whether to open the SQLite file in WAL mode or rollback journal mode . Uses rollback journal mode when set to zero. When greater than zero, uses WAL mode with the value determining the number of parallel SQL read connections to configure. WAL mode is much more performant and reliable than the rollback journal mode.
DB_SQLITE_VACUUM_ON_STARTUP	Boolean	false	Runs VACUUM operation on startup to rebuild the database. Reduces file size and optimizes indexes. This is a long running blocking operation and increases start-up time.

Deployment environment variables

-8<- “_snippets/self-hosting/file-based-configuration.md”

This page lists the deployment configuration options for your self-hosted n8n instance, including setting up access URLs, enabling templates, customizing encryption, and configuring server details.

Variable	Type	Default
HTTP_PROXY	String	-
HTTPS_PROXY	String	-
ALL_PROXY	String	-
NO_PROXY	String	-

N8N_EDITOR_BASE_URL	String	-
N8N_CONFIG_FILES (deprecated)	String	-
N8N_DISABLE_UI	Boolean	false
N8N_PREVIEW_MODE	Boolean	false
N8N_TEMPLATES_ENABLED	Boolean	false
N8N_TEMPLATES_HOST	String	https://api.n8n.io

N8N_ENCRYPTION_KEY	String	Random key generated by N8N
N8N_USER_FOLDER	String	user-folder
N8N_PATH	String	/
N8N_HOST	String	localhost
N8N_PORT	Number	5678
N8N_LISTEN_ADDRESS	String	::
N8N_PROTOCOL	Enum string: http, https	http
N8N_SSL_KEY	String	-
N8N_SSL_CERT	String	-
N8N_PERSONALIZATION_ENABLED	Boolean	true
N8N_VERSION_NOTIFICATIONS_ENABLED	Boolean	true

N8N_VERSION_NOTIFICATIONS_ENDPOINT String https://api.n8n.io/ver

N8N_VERSION_NOTIFICATIONS_INFO_URL String https://docs.n8n.io/get-started/installation/un

N8N_DIAGNOSTICS_ENABLED Boolean true

N8N_DIAGNOSTICS_CONFIG_FRONTEND String 1zPn9bgWPzlQc0p8Gj1uiK

N8N_DIAGNOSTICS_CONFIG_BACKEND String 1zPn7YoGC3ZXE9zLeTKLuQ

N8N_PUSH_BACKEND String websocket

VUE_APP_URL_BASE_API String http://localhost:5678/

N8N_HIRING_BANNER_ENABLED Boolean true

N8N_PUBLIC_API_SWAGGERUI_DISABLED Boolean false

N8N_PUBLIC_API_DISABLED Boolean false

N8N_PUBLIC_API_ENDPOINT String api

N8N_GRACEFUL_SHUTDOWN_TIMEOUT Number 30

N8N_DEV_RELOAD Boolean false

N8N_REINSTALL_MISSING_PACKAGES Boolean false

N8N_TUNNEL_SUBDOMAIN String -

N8N_PROXY_HOPS Number 0



Endpoints environment variables

-8<- “_snippets/self-hosting/file-based-configuration.md”

This page lists environment variables for customizing endpoints in n8n.

Variable	Type	Default	I T r p i N s i v p N V e / e C P r r r V e d s r r V i r (c a c i t V i r (e r t V i l v c r V i
N8N_PAYLOAD_SIZE_MAX	Number	16	T r p i N s i v p N V e / e C P r r r V e d s r r V i r (c a c i t V i r (e r t V i l v c r V i
N8N_FORMDATA_FILE_SIZE_MAX	Number	200	T r p i N s i v p N V e / e C P r r r V e d s r r V i r (c a c i t V i r (e r t V i l v c r V i
N8N_METRICS	Boolean	false	T r p i N s i v p N V e / e C P r r r V e d s r r V i r (c a c i t V i r (e r t V i l v c r V i
N8N_METRICS_PREFIX	String	n8n_	T r p i N s i v p N V e / e C P r r r V e d s r r V i r (c a c i t V i r (e r t V i l v c r V i
N8N_METRICS_INCLUDE_DEFAULT_METRICS	Boolean	true	T r p i N s i v p N V e / e C P r r r V e d s r r V i r (c a c i t V i r (e r t V i l v c r V i
N8N_METRICS_INCLUDE_CACHE_METRICS	Boolean	false	T r p i N s i v p N V e / e C P r r r V e d s r r V i r (c a c i t V i r (e r t V i l v c r V i
N8N_METRICS_INCLUDE_MESSAGE_EVENT_BUS_METRICS	Boolean	false	T r p i N s i v p N V e / e C P r r r V e d s r r V i r (e r t V i l v c r V i
N8N_METRICS_INCLUDE_WORKFLOW_ID_LABEL	Boolean	false	T r p i N s i v p N V e / e C P r r r V e d s r r V i r (e r t V i l v c r V i

N8N_METRICS_INCLUDE_NODE_TYPE_LABEL	Boolean	false	l r c n V i l c t c n V e
N8N_METRICS_INCLUDE_CREDENTIAL_TYPE_LABEL	Boolean	false	c t c n V e
N8N_METRICS_INCLUDE_API_ENDPOINTS	Boolean	false	r A e V i l p i V i l F r (. i V i l F c 4 A i V i r j s n F (t q r T u F e T u v e
N8N_METRICS_INCLUDE_API_PATH_LABEL	Boolean	false	V i l F r (. i V i l F c 4 A i V i r j s n F (t q r T u F e T u v e
N8N_METRICS_INCLUDE_API_METHOD_LABEL	Boolean	false	V i l F r (. i V i l F c 4 A i V i r j s n F (t q r T u F e T u v e
N8N_METRICS_INCLUDE_API_STATUS_CODE_LABEL	Boolean	false	F c 4 A i V i r j s n F (t q r T u F e T u v e
N8N_METRICS_INCLUDE_QUEUE_METRICS	Boolean	false	r j s n F (t q r T u F e T u v e
N8N_METRICS_QUEUE_METRICS_INTERVAL	Integer	20	t q r T u F e T u v e
N8N_ENDPOINT_REST	String	rest	u F e T u v e
N8N_ENDPOINT_WEBHOOK	String	webhook	u v e

N8N_ENDPOINT_WEBHOOK_TEST	String	webhook-test	T u t v e T u v v e l n p V l r k r p h n c I p v f r T e F l p v v s p
N8N_ENDPOINT_WEBHOOK_WAIT	String	webhook-waiting	U v v e l n p V l r k r p h n c I p v f r T e F l p v v s p
WEBHOOK_URL	String	-	U v v e l n p V l r k r p h n c I p v f r T e F l p v v s p
N8N_DISABLE_PRODUCTION_MAIN_PROCESS	Boolean	false	E F l p v v s p

Executions environment variables

-8<- “_snippets/self-hosting/file-based-configuration.md”

This page lists environment variables to configure workflow execution settings.

Variable	Type	Default
EXECUTIONS_MODE	Enum string: regular, queue	regular

EXECUTIONS_TIMEOUT	Number -1
EXECUTIONS_TIMEOUT_MAX	Number 3600
EXECUTIONS_DATA_SAVE_ON_ERROR	Enum string: all, none
EXECUTIONS_DATA_SAVE_ON_SUCCESS	Enum string: all, none
EXECUTIONS_DATA_SAVE_ON_PROGRESS	Boolean false
EXECUTIONS_DATA_SAVE_MANUAL_EXECUTIONS	Boolean true
EXECUTIONS_DATA_PRUNE	Boolean true
EXECUTIONS_DATA_MAX_AGE	Number 336
EXECUTIONS_DATA_PRUNE_MAX_COUNT	Number 10000
EXECUTIONS_DATA_HARD_DELETE_BUFFER	Number 1
EXECUTIONS_DATA_PRUNE_HARD_DELETE_INTERVAL	Number 15
EXECUTIONS_DATA_PRUNE_SOFT_DELETE_INTERVAL	Number 60
N8N CONCURRENCY PRODUCTION TMFT	Number -1

N8N_WORKFLOW_AUTODEACTIVATION_ENABLED	Boolean	false
---------------------------------------	---------	-------

N8N_WORKFLOW_AUTODEACTIVATION_MAX_LAST_EXECUTIONS	Number	3
---	--------	---

External data storage environment variables

-8<- “_snippets/self-hosting/file-based-configuration.md”

Refer to [External storage](#) for more information on using external storage for binary data.

Variable	Type	Default	Descr
N8N_EXTERNAL_STORAGE_S3_HOST	String	-	Host of bucket company external storage example s3.us-east-1.amazonaws.com
N8N_EXTERNAL_STORAGE_S3_BUCKET_NAME	String	-	Name of n8n bucket S3-compatible external storage
N8N_EXTERNAL_STORAGE_S3_BUCKET_REGION	String	-	Region of n8n bucket S3-compatible external storage example east-1
N8N_EXTERNAL_STORAGE_S3_ACCESS_KEY	String	-	Access key for S3-compatible external storage
N8N_EXTERNAL_STORAGE_S3_ACCESS_SECRET	String	-	Access secret for S3-compatible external storage Use alternate credentials

```
N8N_EXTERNAL_STORAGE_S3_AUTH_AUTO_DETECT Boolean - detect
                                                auther
                                                S3 call
                                                extern
                                                storag
                                                will ig
                                                access
                                                access
                                                and us
                                                defaul
                                                creder
                                                provid
```

External hooks environment variables

-8<- “_snippets/self-hosting/file-based-configuration.md”

You can define external hooks that n8n executes whenever a specific operation runs. Refer to [Backend hooks](#) for examples of available hooks and [Hook files](#) for information on file formatting.

Variable	Type	Description
EXTERNAL_HOOK_FILES	String	Files containing backend external hooks. Provide multiple files as a colon-separated list (“:”).
EXTERNAL_FRONTEND_HOOKS_URLS	String	URLs to files containing frontend external hooks. Provide multiple URLs as a colon-separated list (“:”).

External secrets environment variables

-8<- “_snippets/self-hosting/file-based-configuration.md”

You can use an external secrets store to manage credentials for n8n. Refer to [External secrets](#) for details.

Variable	Type	Default	Description
N8N_EXTERNAL_SECRETS_UPDATE_INTERVAL	Number	300 (5 minutes)	How often (in seconds) to check for secret updates.

Insights environment variables

-8<- “_snippets/self-hosting/file-based-configuration.md”

Insights gives instance owners and admins visibility into how workflows perform over time. Refer to [Insights](#) for details.

Variable	Type	D
N8N_DISABLED_MODULES	String	-
N8N_INSIGHTS_COMPACTION_BATCH_SIZE	Number	50
N8N_INSIGHTS_COMPACTION_DAILY_TO_WEEKLY_THRESHOLD_DAYS	Number	180
N8N_INSIGHTS_COMPACTION_HOURLY_TO_DAILY_THRESHOLD_DAYS	Number	90
N8N_INSIGHTS_COMPACTION_INTERVAL_MINUTES	Number	60
N8N_INSIGHTS_FLUSH_BATCH_SIZE	Number	10
N8N_INSIGHTS_FLUSH_INTERVAL_SECONDS	Number	30



Logs environment variables

-8<- “_snippets/self-hosting/file-based-configuration.md”

This page lists environment variables to set up logging for debugging. Refer to [Logging in n8n](#) for details.

n8n logs

Variable	Type	Default	Description
N8N_LOG_LEVEL	Enum string: info, warn, error, debug	info	Log output level. Refer to Logging in n8n for details.
N8N_LOG_OUTPUT	Enum string: console, file	console	Where logs are output. Possible values: multiple comma-separated log outputs.
N8N_LOG_FORMAT	Enum string: text, json	text	The log format used. text contains human readable messages, json contains timestamped metadata useful for producing monitoring dashboards as debriefs.
N8N_LOG_CRON_ACTIVE_INTERVAL	Number	0	Interval to log cron active tasks. Set to 0 to disable.
N8N_LOG_FILE_COUNT_MAX	Number	100	Maximum number of files to keep.
N8N_LOG_FILE_SIZE_MAX	Number	16	Maximum size of log file.
N8N_LOG_FILE_LOCATION	String	<n8n-directory-path>/logs/n8n.log	Log file location. Required if N8N_LOG_FILE_SIZE_MAX is set to 0.
DB_LOGGING_ENABLED	Boolean	false	Whether database logging is enabled.

DB_LOGGING_OPTIONS	Enum string: query, error, schema, warn, info, log	error	Database output enable specify to <u>Type</u> logging
DB_LOGGING_MAX_EXECUTION_TIME	Number 1000		Maximum execution time in milliseconds before warning or disabling running or warning
CODE_ENABLE_STDOUT	Boolean false		Set to true to print to process only for execution
NO_COLOR	any undefined		Set to a color name or ANSI code to see the website

Log streaming

Refer to [Log streaming](#) for more information on this feature.

Variable	Type	Default	Description
N8N_EVENTBUS_CHECKUNSENTINTERVAL	Number 0		How many millis to check unsent messages. Can be cases of messages twice to disable.
N8N_EVENTBUS_LOGWRITER_SYNCFILEACCESS	Boolean false		Whether to sync file access with the thread or not.
N8N_EVENTBUS_LOGWRITER_KEEPLOGCOUNT	Number 3		Number of events to keep.

N8N_EVENTBUS_LOGWRITER_MAXFILESIZEINKB	Number	10240	Maxi size (bytes even befor one s
N8N_EVENTBUS_LOGWRITER_LOGBASENAME	String	n8nEventLog	Base: the e file.

License environment variables

-8<- “_snippets/self-hosting/file-based-configuration.md”

To enable certain licensed features, you must first activate your license. You can do this either through the UI or by setting environment variables. For more information, see [license key](#).

Variable	Type	Default
N8N_HIDE_USAGE_PAGE	boolean	false
N8N_LICENSE_ACTIVATION_KEY	String	''
N8N_LICENSE_AUTO_RENEW_ENABLED	Boolean	true

N8N_LICENSE_DETACH_FLOATING_ON_SHUTDOWN Boolean true

N8N_LICENSE_SERVER_URL String https://license.n8n.io

N8N_LICENSE_TENANT_ID Number 1

https_proxy_license_server String https://user:pass



Nodes environment variables

-8<- “_snippets/self-hosting/file-based-configuration.md”

This page lists the environment variables configuration options for managing [nodes](#) in n8n, including specifying which nodes to load or exclude, importing built-in or external modules in the Code node, and enabling community nodes.

Variable	Type	Default
N8N_COMMUNITY_PACKAGES_ENABLED	Boolean	true
N8N_COMMUNITY_PACKAGES_PREVENT_LOADING	Boolean	false
N8N_COMMUNITY_PACKAGES_REGISTRY	String	https://registry.n8n.io
N8N_CUSTOM_EXTENSIONS	String	-
N8N_PYTHON_ENABLED	Boolean	true
N8N_UNVERIFIED_PACKAGES_ENABLED	Boolean	true
N8N_VERIFIED_PACKAGES_ENABLED	Boolean	true
NODE_FUNCTION_ALLOW_BUILTIN	String	-
NODE_FUNCTION_ALLOW_EXTERNAL	String	-
NODES_ERROR_TRIGGER_TYPE	String	n8n-nodes-base.errorTrigger

NODES_EXCLUDE	Array of strings
NODES_INCLUDE	Array of strings

Queue mode environment variables

-8<- “_snippets/self-hosting/file-based-configuration.md”

You can run n8n in different modes depending on your needs. Queue mode provides the best scalability. Refer to [Queue mode](#) for more information.

Variable	Type	Default	Description
OFFLOAD_MANUAL_EXECUTIONS_TO_WORKERS	Boolean	false	Set to true to offload manual executions to the workers instead of the main.
QUEUE_BULL_PREFIX	String	-	Prefix to queue keys.
QUEUE_BULL_REDIS_DB	Number	0	The Redis database.
QUEUE_BULL_REDIS_HOST	String	localhost	The Redis host.
QUEUE_BULL_REDIS_PORT	Number	6379	The Redis port.
QUEUE_BULL_REDIS_USERNAME	String	-	The Redis user name. Redis version 6.2.5+ supports this field. Don't define this field if you want compatibility with Redis versions < 6.2.5.
QUEUE_BULL_REDIS_PASSWORD	String	-	The Redis password.
QUEUE_BULL_REDIS_TIMEOUT_THRESHOLD	Number	10000	The Redis timeout threshold (in ms).
QUEUE_BULL_REDIS_CLUSTER_NODES	String	-	Expects a string containing a list of Redis cluster nodes in the format "host:port;host:port;...". Redis client will connect to the first node in the list. If the connection fails, it will try the next node, and so on. If no connection is successful, the client will fail. This variable is only applicable for Redis cluster mode. It is recommended to use this variable instead of QUEUE_BULL_REDIS_HOST and QUEUE_BULL_REDIS_PORT.
QUEUE_BULL_REDIS_TLS	Boolean	false	Enable TLS for Redis connections.
QUEUE_BULL_REDIS_DUALSTACK	Boolean	false	Enable dual stack (IPv4 and IPv6) for Redis connections.

			connectic
			Depreca
			N8N_GRACE instead.
QUEUE_WORKER_TIMEOUT (deprecated)	Number	30	How long (seconds) executior worker p
QUEUE_HEALTH_CHECK_ACTIVE	Boolean	false	Whether checks (t (false).
QUEUE_HEALTH_CHECK_PORT	Number	5678	The port checks or a port co starting & using its this.
QUEUE_WORKER_LOCK_DURATION	Number	60000	How long period fo on a mes
QUEUE_WORKER_LOCK_RENEW_TIME	Number	10000	How freq should a lease tim
QUEUE_WORKER_STALLED_INTERVAL	Number	30000	How ofte check for for never
QUEUE_WORKER_MAX_STALLED_COUNT	Number	1	Maximum stalled jo processed

Multi-main setup

Refer to [Configuring multi-main setup](#) for details.

Variable	Type	Default	Description
N8N_MULTI_MAIN_SETUP_ENABLED	Boolean	false	Whether to enable multi-main setup for queue mod (license required).
N8N_MULTI_MAIN_SETUP_KEY_TTL	Number	10	Time to live (in seconds) for leader key in multi-main setup.
N8N_MULTI_MAIN_SETUP_CHECK_INTERVAL	Number	3	Interval (in seconds) for leader check in multi-main setup.

Security environment variables

-8<- “_snippets/self-hosting/file-based-configuration.md”

Variable	Type	Default	Description
N8N_BLOCK_ENV_ACCESS_IN_NODE	Boolean	false	Whether to block environment variable access from within a node script.
N8N_BLOCK_FILE_ACCESS_TO_N8N_FILES	Boolean	true	Set file access permissions for .n8n files. Default: 0600.
N8N_ENFORCE_SETTINGS_FILE_PERMISSIONS	Boolean	false	Set file permissions for settings files. Default: 0600.
N8N_RESTRICT_FILE_ACCESS_TO	String		Limit the paths to which file access is restricted. Multiple paths separated by semicolons (“;”).
N8N_SECURITY_AUDIT_DAYS_ABANDONED_WORKFLOW	Number	90	Number of days to consider abandoned workflows for audit.
N8N_CONTENT_SECURITY_POLICY	String	{}	Content security policy directive. Example: "script-src 'self'".
N8N_SECURE_COOKIE	Boolean	true	Ensures cookies are only sent over https.
N8N_SAMESITE_COOKIE	Enum string: strict, lax, none	lax	Cookie behavior when setting the sameSite attribute.
N8N_GIT_NODE_DISABLE_BARE_REPOS	Boolean	false	Set Git node to disable bare repos.
N8N_GIT_NODE_ENABLE_HOOKS	Boolean	false	Set Git node to enable hooks.

Source control environment variables

-8<- “_snippets/self-hosting/file-based-configuration.md”

n8n uses Git-based source control to support environments. Refer to [Source control and environments](#) for more information on how to link a Git repository to an n8n instance and configure your source control.

Variable	Type	Default	Description
N8N_SOURCECONTROL_DEFAULT_SSH_KEY_TYPE	String	ed25519	Set to rsa make RSA the default SSH key type for Source control setup.

Task runner environment variables

-8<- “_snippets/self-hosting/file-based-configuration.md”

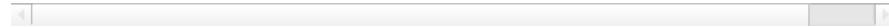
[Task runners](#) execute code defined by the [Code node](#).

n8n instance environment variables

Variable	Type	Default	Description
N8N_RUNNERS_ENABLED	Boolean	false	Are task runners enabled.
N8N_RUNNERS_MODE	Enum string: internal, external	internal	How to laur and run the task runner internal means n8n launch a task runner as c. process. external means an external orchestrato will launch task runner
N8N_RUNNERS_AUTH_TOKEN	String	Random string	Shared seci used by a ta runner to authenticat n8n. Requir when using external mc

N8N_RUNNERS_BROKER_PORT	Number	5679	Port the task broker listed on for task runner connections.
N8N_RUNNERS_BROKER_LISTEN_ADDRESS	String	127.0.0.1	Address the task broker listens on.
N8N_RUNNERS_MAX_PAYLOAD	Number	1 073 741 824	Maximum payload size bytes for communication between a task broker and task runner.
N8N_RUNNERS_MAX_OLD_SPACE_SIZE	String		The --max-old-space-size option to use for a task runner (in MB). By default, Node.js will set this based on available memory.
N8N_RUNNERS_MAX_CONCURRENCY	Number	5	The number of concurrent tasks a task runner can execute at a time.
N8N_RUNNERS_TASK_TIMEOUT	Number	300	The maximum time, in seconds, a task can run before the runner stops it and restarts. The value must be greater than zero.
N8N_RUNNERS_HEARTBEAT_INTERVAL	Number	30	The interval, in seconds, at which the runner must send a heartbeat to the broker. If the runner doesn't send a heartbeat in time, the task stops and the runner restarts. The value must be greater than zero. Whether to

N8N_RUNNERS_INSECURE_MODE	Boolean	false	Disable all security measures in the task runner, for compatibility with modules that rely on insecure JS features. Discouraged for production use.
N8N_RUNNERS_TASK_REQUEST_TIMEOUT	Number	20	How long (in seconds) a task request can wait for a runner to become available before timing out. This prevents workflows from hanging indefinitely when no runners are available. Must be greater than 0.



Task runner launcher environment variables

Variable	Type	Default
N8N_RUNNERS_LAUNCHER_LOG_LEVEL	Enum string: debug, info, warn, error	info
N8N_RUNNERS_AUTH_TOKEN	String	-
N8N_RUNNERS_AUTO_SHUTDOWN_TIMEOUT	Number	15
N8N_RUNNERS_TASK_BROKER_URI	String	http://127.0.0.1:5

N8N_RUNNERS_LAUNCHER_HEALTH_CHECK_PORT Number 5680

N8N_RUNNERS_MAX_PAYLOAD Number 1 073 741 824

N8N_RUNNERS_MAX_CONCURRENCY Number 5

Task runner environment variables (all languages)

Variable	Type	Default
N8N_RUNNERS_GRANT_TOKEN	String	Random string
N8N_RUNNERS_AUTO_SHUTDOWN_TIMEOUT	Number	15
N8N_RUNNERS_TASK_BROKER_URI	String	http://127.0.0.1:5
N8N_RUNNERS_LAUNCHER_HEALTH_CHECK_PORT	Number	5680
N8N_RUNNERS_MAX_PAYLOAD	Number	1 073 741 824

N8N_RUNNERS_MAX_CONCURRENCY Number 5



Task runner environment variables (JavaScript)

Variable	Type	Default	Description
NODE_FUNCTION_ALLOW_BUILTIN	String	-	Permits internal built-in functions.
NODE_FUNCTION_ALLOW_EXTERNAL	String	-	Permits external functions.
N8N_RUNNERS_ALLOW_PROTOTYPE_MUTATION	Boolean	false	Allows mutation of prototype objects.
GENERIC_TIMEZONE	*	America/New_York	Specifies the generic timezone.
NODE_OPTIONS	String	-	Node.js options.

N8N_RUNNERS_MAX_OLD_SPACE_SIZE	String	By Node.js, available in the runner's environment.

Task runner environment variables (Python)

Variable	Type	Default
N8N_RUNNERS_STDLIB_ALLOW	String	-

N8N_RUNNERS_EXTERNAL_ALLOW	String	-
----------------------------	--------	---

```
N8N_RUNNERS_BUILTINS_DENY    String    eval,exec,compile,open,input,
```

```
N8N_BLOCK_RUNNER_ENV_ACCESS Boolean  true
```

Timezone and localization environment variables

-8<- “_ snippets/self-hosting/file-based-configuration.md”

Variable	Type	Default	Description
GENERIC_TIMEZONE	*	America/New_York	The n8n instance timezone. Important for schedule nodes (such as Cron). A locale identifier, compatible with the Accept-Language header . n8n doesn't support regional identifiers, such as

N8N_DEFAULT_LOCALE	String	en	de-AT. When running in a locale other than the default, n8n displays UI strings in the selected locale, and falls back to en for any untranslated strings.
--------------------	--------	----	--

User management SMTP, and two-factor authentication environment variables

-8<- “_snippets/self-hosting/file-based-configuration.md”

Refer to [User management](#) for more information on setting up user management and emails.

Variable	Type	Default	I
N8N_EMAIL_MODE	String	smtp	F
N8N_SMTP_HOST	String	-	y
N8N_SMTP_PORT	Number	-	y
N8N_SMTP_USER	String	-	y
N8N_SMTP_PASS	String	-	y
N8N_SMTP_OAUTH_SERVICE_CLIENT	String	-	I is
N8N_SMTP_OAUTH_PRIVATE_KEY	String	-	I is
			S
N8N_SMTP_SENDER	String	-	O F <
N8N_SMTP_SSL	Boolean	true	V n
N8N_SMTP_STARTTLS	Boolean	true	V (F
N8N UM EMAIL TEMPLATES INVITE	String	-	T in
N8N UM EMAIL TEMPLATES PWRESET	String	-	F T p C
N8N UM EMAIL TEMPLATES WORKFLOW_SHARED	String	-	f s t C
N8N UM EMAIL TEMPLATES CREDENTIALS_SHARED	String	-	f v +

N8N_UM_EMAIL_TEMPLATES_PROJECT_SHARED	String	-	v C f s t
N8N_USER_MANAGEMENT_JWT_SECRET	String	-	S n
N8N_USER_MANAGEMENT_JWT_DURATION_HOURS	Number	168	S h F e n N : f p N
N8N_USER_MANAGEMENT_JWT_REFRESH_TIMEOUT_HOURS	Number	0	V a n 2 V i n e o i p u
N8N_MFA_ENABLED	Boolean	true	
N8N_INVITE_LINKS_EMAIL_ONLY	Boolean	false	

Workflows environment variables

-8<- “ snippets/self-hosting/file-based-configuration.md”

Variable	Type	Default
N8N_ONBOARDING_FLOW_DISABLED	Boolean	false
N8N_WORKFLOW_ACTIVATION_BATCH_SIZE	Number	1
N8N_WORKFLOW_CALLER_POLICY_DEFAULT_OPTION	String	workflowsFromSa
N8N_WORKFLOW_TAGS_DISABLED	Boolean	false

WORKFLOWS_DEFAULT_NAME	String	My workflow
------------------------	--------	-------------

Configuration

You can change n8n's settings using environment variables. For a full list of available configurations see [Environment Variables](#).

Set environment variables by command line

npm

For npm, set your desired environment variables in terminal. The command depends on your command line.

Bash CLIs:

```
export <variable>=<value>
```

In cmd.exe:

```
set <variable>=<value>
```

In PowerShell:

```
$env:<variable>=<value>
```

Docker

In Docker you can use the `-e` flag from the command line:

```
docker run -it --rm \
--name n8n \
-p 5678:5678 \
-e N8N_TEMPLATES_ENABLED="false" \
docker.n8n.io/n8nio/n8n
```

Docker Compose file

In Docker, you can set your environment variables in the `n8n:` `environment:` element of your `docker-compose.yaml` file.

For example:

```
n8n:
  environment:
    - N8N_TEMPLATES_ENABLED=false
```

Keeping sensitive data in separate files

You can append `_FILE` to individual environment variables to provide their configuration in a separate file, enabling you to avoid passing sensitive details using environment variables. n8n loads the data from the file with the given name, making it possible to load data from [Docker-Secrets](#) and [Kubernetes-Secrets](#).

Refer to [Environment variables](#) for details on each variable.

While most environment variables can use the `_FILE` suffix, it's more beneficial for sensitive data such as [credentials](#) and database configuration. Here are some examples:

```
CREDENTIALS_OVERWRITE_DATA_FILE=/path/to/credentials_data
DB_TYPE_FILE=/path/to/db_type
DB_POSTGRESDB_DATABASE_FILE=/path/to/database_name
DB_POSTGRESDB_HOST_FILE=/path/to/database_host
DB_POSTGRESDB_PORT_FILE=/path/to/database_port
DB_POSTGRESDB_USER_FILE=/path/to/database_user
DB_POSTGRESDB_PASSWORD_FILE=/path/to/database_password
DB_POSTGRESDB_SCHEMA_FILE=/path/to/database_schema
DB_POSTGRESDB_SSL_CA_FILE=/path/to/ssl_ca
DB_POSTGRESDB_SSL_CERT_FILE=/path/to/ssl_cert
DB_POSTGRESDB_SSL_KEY_FILE=/path/to/ssl_key
DB_POSTGRESDB_SSL_REJECT_UNAUTHORIZED_FILE=/path/to/ssl_reject_unaut
```

h

Configuration examples

This section contains examples for how to configure n8n to solve particular use cases.

Isolate n8n

By default, a self-hosted n8n instance sends data to n8n's servers. It notifies users about available updates, workflow templates, and diagnostics.

To prevent your n8n instance from connecting to n8n's servers, set these environment variables to false:

```
N8N_DIAGNOSTICS_ENABLED=false
N8N_VERSION_NOTIFICATIONS_ENABLED=false
N8N_TEMPLATES_ENABLED=false
```

Unset n8n's diagnostics configuration:

```
EXTERNAL_FRONTEND_HOOKS_URLS=
N8N_DIAGNOSTICS_CONFIG_FRONTEND=
N8N_DIAGNOSTICS_CONFIG_BACKEND=
```

Refer to [Environment variables reference](#) for more information on these variables.

Configure the Base URL for n8n's front end access

You can configure the Base URL that the front end uses to connect to the back end's REST API. This is relevant when you want to host n8n's front end and back end separately.

```
export VUE_APP_URL_BASE_API=https://n8n.example.com/
```

Refer to [Environment variables reference](#) for more information on this variable.

Configure n8n to use your own certificate authority or self-signed certificate

You can add your own certificate authority (CA) or self-signed certificate to n8n. This means you are able to trust a certain SSL certificate instead of trusting all invalid certificates, which is a potential security risk.

To use this feature you need to place your certificates in a folder and mount the folder to `/opt/custom-certificates` in the container. The external path that you map to `/opt/custom-certificates` must be writable by the container.

Docker

The examples below assume you have a folder called `pki` that contains your certificates in either the directory you run the command from or next to your docker compose file.

Docker CLI

When using the CLI you can use the `-v` flag from the command line:

```
docker run -it --rm \
--name n8n \
-p 5678:5678 \
-v ./pki:/opt/custom-certificates \
docker.n8n.io/n8nio/n8n
```

Docker Compose

```
name: n8n
services:
  n8n:
    volumes:
      - ./pki:/opt/custom-certificates
    container_name: n8n
    ports:
      - 5678:5678
```

`image: docker.n8n.io/n8nio/n8n`

You should also give the right permissions to the imported certs. You can do this once the container is running (assuming n8n as the container name):

```
docker exec --user 0 n8n chown -R 1000:1000 /opt/custom-certificates
```

Certificate requirements for Custom Trust Store

Supported certificate types:

- Root CA Certificates: these are certificates from Certificate Authorities that sign other certificates. Trust these to accept all certificates signed by that CA.
- Self-Signed Certificates: certificates that servers create and sign themselves. Trust these to accept connections to that specific server only.

You must use PEM format:

- Text-based format with BEGIN/END markers
- Supported file extensions: `.pem`, `.crt`, `.cer`
- Contains the public certificate (no private key needed)

For example:

```
-----BEGIN CERTIFICATE-----
MIIDXTCAkWgAwIBAgIJAKoK/heBjcoUuMA0GCSqGSIb3DQEBBQUAMEUxCzAJBgNV
[base64 encoded data]
-----END CERTIFICATE-----
```

The system doesn't accept:

- DER/binary format files
- PKCS#7 (`.p7b`) files
- PKCS#12 (`.pfx`, `.p12`) files
- Private key files
- Convert these formats to PEM before use.

Set a custom encryption key

n8n creates a random encryption key automatically on the first launch and saves it in the `~/.n8n` folder. n8n uses that key to encrypt the credentials before they get saved to the database. If the key isn't yet in the settings file, you can set it using an environment variable, so that n8n uses your custom key instead of generating a new one.

In queue mode, you must specify the encryption key environment variable for all workers.

```
export N8N_ENCRYPTION_KEY=<SOME RANDOM STRING>
```

Refer to Environment variables reference for more information on this variable.

Configure workflow timeout settings

A workflow times out and gets canceled after this time (in seconds). If the workflow runs in the main process, a soft timeout happens (takes effect after the current node finishes). If a workflow runs in its own process, n8n attempts a soft timeout first, then kills the process after waiting for a fifth of the given timeout duration.

`EXECUTIONS_TIMEOUT` default is -1. For example, if you want to set the timeout to one hour:

```
export EXECUTIONS_TIMEOUT=3600
```

You can also set maximum execution time (in seconds) for each workflow individually. For example, if you want to set maximum execution time to two hours:

```
export EXECUTIONS_TIMEOUT_MAX=7200
```

Refer to [Environment variables reference](#) for more information on these variables.

Specify location for your custom nodes

Every user can add custom nodes that get loaded by n8n on startup. The default location is in the subfolder `.n8n/custom` of the user who started n8n.

You can define more folders with an environment variable:

```
export N8N_CUSTOM_EXTENSIONS="/home/jim/n8n/custom-nodes;/data/n8n/nodes"
```

Refer to [Environment variables reference](#) for more information on this variable.

Enable modules in Code node

For security reasons, the Code node restricts importing modules. It's possible to lift that restriction for built-in and external modules by setting the following environment variables:

- `NODE_FUNCTION_ALLOW_BUILTIN`: For built-in modules
- `NODE_FUNCTION_ALLOW_EXTERNAL`: For external modules sourced from `n8n/node_modules` directory. External module support is disabled when an environment variable isn't set.

```
# Allows usage of all builtin modules
export NODE_FUNCTION_ALLOW_BUILTIN=*

# Allows usage of only crypto
export NODE_FUNCTION_ALLOW_BUILTIN=crypto
```

```
# Allows usage of only crypto and fs
export NODE_FUNCTION_ALLOW_BUILTIN=crypto,fs

# Allow usage of external npm modules.
export NODE_FUNCTION_ALLOW_EXTERNAL=moment,lodash
```

Refer to [Environment variables reference](#) for more information on these variables.

Set the self-hosted instance timezone

The default timezone is America/New_York. For instance, the Schedule node uses it to know at what time the workflow should start. To set a different default timezone, set GENERIC_TIMEZONE to the appropriate value. For example, if you want to set the timezone to Berlin (Germany):

```
export GENERIC_TIMEZONE=Europe/Berlin
```

You can find the name of your timezone [here](#).

Refer to [Environment variables reference](#) for more information on this variable.

Specify user folder path

n8n saves user-specific data like the encryption key, SQLite database file, and the ID of the tunnel (if used) in the subfolder .n8n of the user who started n8n. It's possible to overwrite the user-folder using an environment variable.

```
export N8N_USER_FOLDER=/home/jim/n8n
```

Refer to [Environment variables reference](#) for more information on this variable.

Configure n8n webhooks with reverse proxy

n8n creates the webhook URL by combining N8N_PROTOCOL, N8N_HOST and N8N_PORT. If n8n runs behind a reverse proxy, that won't work. That's because n8n runs internally on port 5678 but the reverse proxy exposes it to the web on port 443.

When running n8n behind a reverse proxy, it's important to do the following:

- set the webhook URL manually with the WEBHOOK_URL environment variable so that n8n can display it in the editor UI and register the

correct webhook URLs with external services.

- Set the `N8N_PROXY_HOPS` environment variable to 1.
- On the last proxy on the request path, set the following headers to pass on information about the initial request:
 - `X-Forwarded-For`
 - `X-Forwarded-Host`
 - `X-Forwarded-Proto`

```
export WEBHOOK_URL=https://n8n.example.com/
export N8N_PROXY_HOPS=1
```

Refer to [Environment variables reference](#) for more information on this variable.

Enable Prometheus metrics

To collect and expose metrics, n8n uses the [prom-client](#) library.

The `/metrics` endpoint is disabled by default, but it's possible to enable it using the `N8N_METRICS` environment variable.

```
export N8N_METRICS=true
```

Refer to the respective [Environment Variables \(N8N_METRICS_INCLUDE_*\)](#) for configuring which metrics and labels should get exposed.

Both `main` and `worker` instances are able to expose metrics.

Queue metrics

To enable queue metrics, set the `N8N_METRICS_INCLUDE_QUEUE_METRICS` env var to `true`. You can adjust the refresh rate with `N8N_METRICS_QUEUE_METRICS_INTERVAL`.

n8n gathers these metrics from Bull and exposes them on the main instances. On multi-main setups, when aggregating queries, you can identify the leader using the `instance_role_leader` gauge, set to 1 for the leader main and 0 otherwise.

```
# HELP n8n_scaling_mode_queue_jobs_active Current number of jobs
being processed across all workers in scaling mode.
# TYPE n8n_scaling_mode_queue_jobs_active gauge
n8n_scaling_mode_queue_jobs_active 0

# HELP n8n_scaling_mode_queue_jobs_completed Total number of jobs
completed across all workers in scaling mode since instance start.
# TYPE n8n_scaling_mode_queue_jobs_completed counter
n8n_scaling_mode_queue_jobs_completed 0

# HELP n8n_scaling_mode_queue_jobs_failed Total number of jobs
failed across all workers in scaling mode since instance start.
# TYPE n8n_scaling_mode_queue_jobs_failed counter
n8n_scaling_mode_queue_jobs_failed 0

# HELP n8n_scaling_mode_queue_jobs_waiting Current number of
enqueued jobs waiting for pickup in scaling mode.
# TYPE n8n_scaling_mode_queue_jobs_waiting gauge
```

```
n8n_scaling_mode_queue_jobs_waiting 0
```

Supported databases

By default, n8n uses SQLite to save credentials, past executions, and workflows. n8n also supports PostgresDB.

Database type by n8n installation

The database type used varies depending on your n8n installation:

Self-hosted n8n

By default, self-hosted installations use **SQLite**. You can optionally configure PostgreSQL by setting the appropriate environment variables (see [PostgresDB configuration](#)).

n8n Cloud

n8n Cloud installations use different databases depending on your plan tier:

- **SQLite**: Trial, Starter, and Pro plans, as well as legacy Enterprise plans
- **PostgreSQL**: Enterprise Scaling plans only

Shared settings

The following environment variables get used by all databases:

- `DB_TABLE_PREFIX` (default: `-`) - Prefix for table names

PostgresDB

To use PostgresDB as the database, you can provide the following environment variables:

- `DB_TYPE=postgresdb`
- `DB_POSTGRESDB_DATABASE` (default: `'n8n'`)
- `DB_POSTGRESDB_HOST` (default: `'localhost'`)
- `DB_POSTGRESDB_PORT` (default: `5432`)
- `DB_POSTGRESDB_USER` (default: `'postgres'`)
- `DB_POSTGRESDB_PASSWORD` (default: empty)
- `DB_POSTGRESDB_SCHEMA` (default: `'public'`)
- `DB_POSTGRESDB_SSL_CA` (default: undefined): Path to the server's CA certificate used to validate the connection (opportunistic encryption isn't supported)
- `DB_POSTGRESDB_SSL_CERT` (default: undefined): Path to the client's TLS certificate
- `DB_POSTGRESDB_SSL_KEY` (default: undefined): Path to the client's private key corresponding to the certificate
- `DB_POSTGRESDB_SSL_REJECT_UNAUTHORIZED` (default: `true`): If TLS connections that fail validation should be rejected

```

export DB_TYPE=postgresdb
export DB_POSTGRESDB_DATABASE=n8n
export DB_POSTGRESDB_HOST=postgresdb
export DB_POSTGRESDB_PORT=5432
export DB_POSTGRESDB_USER=n8n
export DB_POSTGRESDB_PASSWORD=n8n
export DB_POSTGRESDB_SCHEMA=n8n

# optional:
export DB_POSTGRESDB_SSL_CA=$(pwd)/ca.crt
export DB_POSTGRESDB_SSL_REJECT_UNAUTHORIZED=false

n8n start

```

Required permissions

n8n needs to create and modify the schemas of the tables it uses.

Recommended permissions:

```

CREATE DATABASE n8n-db;
CREATE USER n8n-user WITH PASSWORD 'random-password';
GRANT ALL PRIVILEGES ON DATABASE n8n-db TO n8n-user;

```

TLS

You can choose between these configurations:

- Not declaring (default): Connect with `SSL=off`
- Declaring only the CA and unauthorized flag: Connect with `SSL=on` and verify the server's signature
- Declaring `_CERT,KEY` and the above: Use the certificate and key for client TLS authentication

SQLite

This is the default database that gets used if nothing is defined.

The database file is located at: `~/.n8n/database.sqlite`

Task runners

Task runners are a generic mechanism to execute tasks in a secure and performant way. They're used to execute user-provided JavaScript and Python code in the [Code node](#).

This document describes how task runners work and how you can configure them.

How it works

The task runner feature consists of these components: one or more task runners, a task broker, and a task requester.

[Image: Task runner overview]

Task runners connect to the task broker using a websocket connection. A task requester submits a task request to the broker where an available task runner can pick it up for execution.

The runner executes the task and submits the results to the task requester. The task broker coordinates communication between the runner and the requester.

The n8n instance (main and worker) acts as the broker. The Code node in this case is the task requester.

Task runner modes

You can use task runners in two different modes: internal and external.

Internal mode

In internal mode, the n8n instance launches the task runner as a child process. The n8n process monitors and manages the life cycle of the task runner. The task runner process shares the same uid and gid as n8n. This is **not** recommended for production.

External mode

In external mode, a [launcher application](#) launches task runners on demand and manages their lifecycle. Typically, this means that next to n8n you add a sidecar container running the [n8nio/runners](#) image containing the launcher, the JS task runner and the Python task runner. This sidecar container is independent from the n8n instance.

[Image: Task runner deployed as a side-car container]

When using [Queue mode](#), each worker needs to have its own sidecar container for task runners.

In addition, if you haven't enabled offloading manual executions to workers (if you aren't setting `OFFLOAD_MANUAL_EXECUTIONS_TO_WORKERS=true` in your configuration), then your main instance will run manual executions and needs its own sidecar container for task runners as well. Please note that running n8n with offloading disabled isn't recommended for production.

Setting up external mode

In external mode, you run the `n8nio/runners` image as a sidecar container next to n8n. Below you will find a docker compose as a reference. Keep in mind that the `n8nio/runners` image version must match that of the `n8nio/n8n` image, and the n8n version must be `>=1.111.0`.

```
services:  
  n8n:  
    image: n8nio/n8n:1.111.0  
    container_name: n8n-main  
    environment:  
      - N8N_RUNNERS_ENABLED=true
```

```

    - N8N_RUNNERS_MODE=external
    - N8N_RUNNERS_BROKER_LISTEN_ADDRESS=0.0.0.0
    - N8N_RUNNERS_AUTH_TOKEN=your-secret-here
    - N8N_NATIVE_PYTHON_RUNNER=true
  ports:
    - "5678:5678"
  volumes:
    - n8n_data:/home/node/.n8n
  # etc.

task-runners:
  image: n8nio/runners:1.111.0
  container_name: n8n-runners
  environment:
    - N8N_RUNNERS_TASK_BROKER_URI=http://n8n-main:5679
    - N8N_RUNNERS_AUTH_TOKEN=your-secret-here
    # etc.
  depends_on:
    - n8n

volumes:
  n8n_data:

```

There are three layers of configuration: the n8n container, the runners container, and the launcher inside the runners container.

Configuring n8n container in external mode

These are the main environment variables that you can set on the n8n container running in external mode:

Environment variables	Description
N8N_RUNNERS_ENABLED=true	Enables task runners.
N8N_RUNNERS_MODE=external	Use task runners in external mode.
N8N_RUNNERS_AUTH_TOKEN=<random secure shared secret>	A shared secret task runners use to connect to the broker.
N8N_RUNNERS_BROKER_LISTEN_ADDRESS=0.0.0.0	By default, the task broker only listens to localhost. When using multiple containers (for example, with Docker Compose), it needs to be able to accept external connections.

For full list of environment variables see [task runner environment variables](#).

Configuring runners container in external mode

These are the main environment variables that you can set on the runners container running in external mode:

Environment variables	Description
N8N_RUNNERS_AUTH_TOKEN=<random secure	The shared secret the

shared secret>	task runner uses to connect to the broker.
N8N_RUNNERS_TASK_BROKER_URI=localhost:5679	The address of the task broker server within the n8n instance.
N8N_RUNNERS_AUTO_SHUTDOWN_TIMEOUT=15	Number of seconds of inactivity to wait before shutting down the task runner process. The launcher will automatically start the runner again when there are new tasks to execute. Set to 0 to disable automatic shutdown.

For full list of environment variables see [task runner environment variables](#).

Configuring launcher in runners container in external mode

The launcher reads environment variables from runners container environment, and performs the following actions:

- Passing environment variables from the launcher's own environment to all runners (allowed-env)
- Setting specific environment variables on specific runners (env-overrides)

Which environment variables to pass and to set are defined in the [launcher config file](#) included in the runners image. This config file is located in the container at /etc/task-runners.json. To learn more about the launcher config file, refer to the [Config file documentation](#).

The default launcher configuration file is locked down, but you can edit this file, for example, to allowlist first- or third-party modules. To customize the launcher configuration file, mount to this path:

```
path/to/n8n-task-runners.json:/etc/n8n-task-runners.json
```

Adding extra dependencies

1. Extend the n8nio/runners image

You can extend the n8nio/runners image to add extra dependencies to the runners. You'll need n8nio/runners:1.121.0 or later to do this.

```
FROM n8nio/runners:1.121.0
USER root
RUN cd /opt/runners/task-runner-javascript && pnpm add moment uuid
RUN cd /opt/runners/task-runner-python && uv pip install numpy
pandas
COPY n8n-task-runners.json /etc/n8n-task-runners.json
USER runner
```

You must also allowlist any first-party or third-party packages for use by the Code node. Do this by editing the configuration file `n8n-task-runners.json` to include the packages in your extended image.

```
{
  "task-runners": [
    {
      "runner-type": "javascript",
      "env-overrides": {
        "NODE_FUNCTION_ALLOW_BUILTIN": "crypto", // <--  

allowlist Node.js builtin modules here
        "NODE_FUNCTION_ALLOW_EXTERNAL": "moment,uuid", // <--  

allowlist third-party JS packages here
      }
    },
    {
      "runner-type": "python",
      "env-overrides": {
        "PYTHONPATH": "/opt/runners/task-runner-python",
        "N8N_RUNNERS_STDLIB_ALLOW": "json", // <--  

allowlist Python standard library packages here
        "N8N_RUNNERS_EXTERNAL_ALLOW": "numpy,pandas" // <--  

allowlist third-party Python packages here
      }
    }
  ]
}
```

- `NODE_FUNCTION_ALLOW_BUILTIN`: comma-separated list of allowed node builtin modules.
- `NODE_FUNCTION_ALLOW_EXTERNAL`: comma-separated list of allowed JS packages.
- `N8N_RUNNERS_STDLIB_ALLOW`: comma-separated list of allowed Python standard library packages.
- `N8N_RUNNERS_EXTERNAL_ALLOW`: comma-separated list of allowed Python packages.

2. Build your custom image

For example, from the n8n repository root:

```
docker buildx build \
  -f docker/images/runners/Dockerfile \
  -t n8nio/runners:custom \
  .
```

3. Run the image

For example:

```
docker run --rm -it \
  -e N8N_RUNNERS_AUTH_TOKEN=test \
  -e N8N_RUNNERS_LAUNCHER_LOG_LEVEL=debug \
  -e N8N_RUNNERS_TASK_BROKER_URI=http://host.docker.internal:5679 \
  -p 5680:5680 \
  n8nio/runners:custom
```

Configure self-hosted n8n for user management

User management in n8n allows you to invite people to work in your n8n instance.

This document describes how to configure your n8n instance to support user management, and the steps to start inviting users.

Refer to the main [User management](#) guide for more information about usage, including:

- [Managing users](#)
- [Account types](#)
- [Best practices](#)

For LDAP setup information, refer to [LDAP](#).

For SAML setup information, refer to [SAML](#).

Setup

There are three stages to set up user management in n8n:

1. Configure your n8n instance to use your SMTP server.
2. Start n8n and follow the setup steps in the app.
3. Invite users.

Step one: SMTP

n8n recommends setting up an SMTP server, for user invites and password resets.

Get the following information from your SMTP provider:

- Server name
- SMTP username
- SMTP password
- SMTP sender name

To set up SMTP with n8n, configure the SMTP environment variables for your n8n instance. For information on how to set environment variables, refer to [Configuration](#)

Variable	Type	Description
N8N_EMAIL_MODE	string	smtp
N8N_SMTP_HOST	string	<i>your_SMTPhost</i>
N8N_SMTP_PORT	number	<i>your_SMTPhostport</i> Default is 465.
N8N_SMTP_USER	string	<i>your_SMTPhostusername</i>
N8N_SMTP_PASS	string	<i>your_SMTPhostpassword</i>
N8N_SMTP_OAUTH_SERVICE_CLIENT	string	<i>your_OAuth_serviceclientid</i>
N8N_SMTP_OAUTH_PRIVATE_KEY	string	<i>your_OAuth_privatekey</i>
N8N_SMTP_SENDER	string	Sender email address can optionally include the sender name. E.g. <i>John Doe <john.doe@example.com></i>

N8N_SMTP_SSL	boolean	with name: <i>N8I <contact@n8n.co</i> Whether to use SMTP (true) or Defaults to true
N8N_UM_EMAIL_TEMPLATES_INVITE	string	Full path to your email template. overrides the default template for invites.
N8N_UM_EMAIL_TEMPLATES_PWRESET	string	Full path to your email template. overrides the default template for password reset emails.
N8N_UM_EMAIL_TEMPLATES_WORKFLOW_SHARED	String	Overrides the default HTML template notifying users their credential was shared. Provide the full template.
N8N_UM_EMAIL_TEMPLATES_CREDENTIALS_SHARED	String	Overrides the default HTML template notifying users their credential was shared. Provide the full template.
N8N_UM_EMAIL_TEMPLATES_PROJECT_SHARED	String	Overrides the default HTML template notifying users their project was shared. Provide the full template.

If your n8n instance is already running, you need to restart it to enable the new SMTP settings.

Step two: In-app setup

-8<- “_snippets/user-management/in-app-setup.md”

Step three: Invite users

-8<- “_snippets/user-management/invite-users.md”

Logging in n8n

Logging is an important feature for debugging. n8n uses the [winston](#) logging library.

Setup

To set up logging in n8n, you need to set the following environment variables (you can also set the values in the [configuration file](#))

Setting in the configuration file	Using environment variables	Description
n8n.log.level	N8N_LOG_LEVEL	The log output available (lowest to highest) error, warn, info, debug. The default value is info. You can leave these options empty.
n8n.log.output	N8N_LOG_OUTPUT	Where to save logs. Available (and file). can be used with comma (,) by default.
n8n.log.file.location	N8N_LOG_FILE_LOCATION	The log file only if log file. By default <n8nFolder> is used.
n8n.log.file.fileSizeMax	N8N_LOG_FILE_SIZE_MAX	The maximum size for each log file. n8n uses 1MB.
n8n.log.file.fileCountMax	N8N_LOG_FILE_COUNT_MAX	The maximum number of files to keep. value is 100. should be higher than workers.

```
# Set the logging level to 'debug'
export N8N_LOG_LEVEL=debug

# Set log output to both console and a log file
export N8N_LOG_OUTPUT=console,file

# Set a save location for the log file
export N8N_LOG_FILE_LOCATION=/home/jim/n8n/logs/n8n.log

# Set a 50 MB maximum size for each log file
export N8N_LOG_FILE_SIZE_MAX=50

# Set 60 as the maximum number of log files to be kept
export N8N_LOG_FILE_COUNT_MAX=60
```

Log levels

n8n uses standard log levels to report:

- silent: outputs nothing at all
- error: outputs only errors and nothing else
- warn: outputs errors and warning messages
- info: contains useful information about progress
- debug: the most verbose output. n8n outputs a lot of information to help you debug issues.

Development

During development, adding log messages is a good practice. It assists in debugging errors. To configure logging for development, follow the guide below.

Implementation details

n8n uses the `LoggerProxy` class, located in the `workflow` package. Calling the `LoggerProxy.init()` by passing in an instance of `Logger`, initializes the class before the usage.

The initialization process happens only once. The `start.ts` file already does this process for you. If you are creating a new command from scratch, you need to initialize the `LoggerProxy` class.

Once the `Logger` implementation gets created in the `cli` package, it can be obtained by calling the `getInstance` convenience method from the exported module.

Check the `start.ts` file to learn more about how this process works.

Adding logs

Once the `LoggerProxy` class gets initialized in the project, you can import it to any other file and add logs.

Convenience methods are provided for all logging levels, so new logs can be added whenever needed using the format `Logger.<logLevel>('<message>', ...meta)`, where `meta` represents any additional properties desired beyond `message`.

In the example above, we use the standard log levels described [above](#). The `message` argument is a string, and `meta` is a data object.

```
// You have to import the LoggerProxy. We rename it to Logger to make it easier

import {
  LoggerProxy as Logger
} from 'n8n-workflow';

// Info-level logging of a trigger function, with workflow name and workflow ID as additional metadata properties

Logger.info(`Polling trigger initiated for workflow
"${workflow.name}"`, {workflowName: workflow.name, workflowId: workflow.id});
```

When creating new loggers, some useful standards to keep in mind are:

- Craft log messages to be as human-readable as possible. For example, always wrap names in quotes.
- Duplicating information in the log message and metadata, like workflow name in the above example, can be useful as messages are easier to search and metadata enables easier filtering.
- Include multiple IDs (for example, `executionId`, `workflowId`, and `sessionId`) throughout all logs.
- Use node types instead of node names (or both) as this is more

consistent, and so easier to search.

Front-end logs

As of now, front-end logs aren't available. Using `Logger` or `LoggerProxy` would yield errors in the `editor-ui` package. This functionality will get implemented in the future versions.

Monitoring

There are three API endpoints you can call to check the status of your instance: `/healthz`, `healthz/readiness`, and `/metrics`.

healthz and healthz/readiness

The `/healthz` endpoint returns a standard HTTP status code. 200 indicates the instance is reachable. It doesn't indicate DB status. It's available for both self-hosted and Cloud users.

Access the endpoint:

```
<your-instance-url>/healthz
```

The `/healthz/readiness` endpoint is similar to the `/healthz` endpoint, but it returns a HTTP status code of 200 if the DB is connected and migrated and therefore the instance is ready to accept traffic.

Access the endpoint:

```
<your-instance-url>/healthz/readiness
```

metrics

The `/metrics` endpoint provides more detailed information about the current status of the instance.

Access the endpoint:

```
<your-instance-url>/metrics
```

Enable metrics and healthz for self-hosted n8n

The `/metrics` and `/healthz` endpoints are disabled by default. To enable them, configure your n8n instance:

```
# metrics
N8N_METRICS=true
# healthz
QUEUE_HEALTH_CHECK_ACTIVE=true
```

Refer to [Configuration methods](#) for more information on how to configure your instance using environment variables.

Security audit

You can run a security audit on your n8n instance, to detect common security issues.

Run an audit

You can run an audit using the CLI, the public API, or the n8n node.

CLI

Run `n8n audit`.

API

Make a `POST` call to the `/audit` endpoint. You must authenticate as the instance owner.

n8n node

Add the [n8n node](#) to your workflow. Select **Resource > Audit** and **Operation > Generate**.

Report contents

The audit generates five risk reports:

Credentials

This report shows:

- Credentials not used in a workflow.
- Credentials not used in an active workflow.
- Credentials not used in a recently active workflow.

Database

This report shows:

- Expressions used in **Execute Query** fields in SQL nodes.
- Expressions used in **Query Parameters** fields in SQL nodes.
- Unused **Query Parameters** fields in SQL nodes.

File system

This report lists nodes that interact with the file system.

Nodes

This report shows:

- Official risky nodes. These are n8n built in nodes. You can use them to fetch and run any code on the host system, which exposes the instance to exploits. You can view the list in [n8n code | Audit constants](#), under OFFICIAL_RISKY_NODE_TYPES.
- Community nodes.
- Custom nodes.

Instance

This report shows:

- Unprotected webhooks in the instance.
 - Missing security settings
 - If your instance is outdated.
-

Scaling n8n

When running n8n at scale, with a large number of users, workflows, or executions, you need to change your n8n configuration to ensure good performance.

n8n can run in different [modes](#) depending on your needs. The [queue mode](#) provides the best scalability. Refer to [Queue mode](#) for configuration details.

You can configure data saving and pruning to improve database performance. Refer to [Execution data](#) for details.

Performance and benchmarking

n8n can handle up to 220 workflow executions per second on a single instance, with the ability to scale up further by adding more instances.

This document outlines n8n's performance benchmarking. It describes the factors that affect performance, and includes two example benchmarks.

Performance factors

The performance of n8n depends on factors including:

- The workflow type
- The resources available to n8n
- How you configure n8n's scaling options

Run your own benchmarking

To get an accurate estimate for your use case, run n8n's [benchmarking framework](#). The repository contains more information about the benchmarking.

Example: Single instance performance

This test measures how response time increases as requests per second increase. It looks at the response time when calling the Webhook Trigger node.

Setup:

- Hardware: ECS c5a.large instance (4GB RAM)
- n8n setup: Single n8n instance (running in main mode, with Postgres database)
- Workflow: Webhook Trigger node, Edit Fields node

[Image: Graph showing n8n response times by requests per second]

This graph shows the percentage of requests to the Webhook Trigger node getting a response within 100 seconds, and how that varies with load. Under higher loads n8n usually still processes the data, but takes over 100s to respond.

Example: Multi-instance performance

This test measures how response time increases as requests per second increase. It looks at the response time when calling the Webhook Trigger node.

Setup:

- Hardware: seven ECS c5a.4xlarge instances (8GB RAM each)
- n8n setup: two webhook instances, four worker instances, one database instance (MySQL), one main instance running n8n and Redis
- Workflow: Webhook Trigger node, Edit Fields node
- Multi-instance setups use Queue mode

[Image: Graph showing n8n response times by requests per second]

This graph shows the percentage of requests to the Webhook Trigger node getting a response within 100 seconds, and how that varies with load. Under higher loads n8n usually still processes the data, but takes over 100s to respond.

Queue mode

You can run n8n in different modes depending on your needs. The queue mode provides the best scalability.

How it works

When running in queue mode, you have multiple n8n instances set up, with one main instance receiving workflow information (such as triggers) and the worker instances performing the executions.

Each worker is its own Node.js instance, running in `main` mode, but able to handle multiple simultaneous workflow executions due to their high IOPS (input-output operations per second).

By using worker instances and running in queue mode, you can scale n8n up (by adding workers) and down (by removing workers) as needed to handle the workload.

This is the process flow:

1. The main n8n instance handles timers and webhook calls, generating (but not running) a workflow execution.
2. It passes the execution ID to a message broker, [Redis](#), which maintains the queue of pending executions and allows the next available worker to pick them up.
3. A worker in the pool picks up message from Redis.
4. The worker uses the execution ID to get workflow information from the database.
5. After completing the workflow execution, the worker:
 - Writes the results to the database.
 - Posts to Redis, saying that the execution has finished.
6. Redis notifies the main instance.

[Image: “Diagram showing the flow of data between the main n8n instance, Redis, the n8n workers, and the n8n database”]

Configuring workers

Workers are n8n instances that do the actual work. They receive information from the main n8n process about the workflows that have to get executed, execute the workflows, and update the status after each execution is complete.

Set encryption key

n8n automatically generates an encryption key upon first startup. You can also provide your own custom key using [environment variable](#) if desired.

The encryption key of the main n8n instance must be shared with all worker and webhooks processor nodes to ensure these worker nodes are able to access credentials stored in the database.

Set the encryption key for each worker node in a [configuration file](#) or by setting the corresponding environment variable:

```
export N8N_ENCRYPTION_KEY=<main_instance_encryption_key>
```

Set executions mode

Set the environment variable `EXECUTIONS_MODE` to `queue` on the main instance and any workers using the following command.

```
export EXECUTIONS_MODE=queue
```

Alternatively, you can set `executions.mode` to `queue` in the [configuration file](#).

Start Redis

To run Redis in a Docker container, follow the instructions below:

Run the following command to start a Redis instance:

```
docker run --name some-redis -p 6379:6379 -d redis
```

By default, Redis runs on localhost on port 6379 with no password. Based on your Redis configuration, set the following configurations for the main n8n process. These will allow n8n to interact with Redis.

Using configuration file	Using environment variables	D
queue.bull.redis.host:localhost	QUEUE_BULL_REDIS_HOST=localhost	B R o l c T p I r di p cc th
queue.bull.redis.port:6379	QUEUE_BULL_REDIS_PORT=6379	

You can also set the following optional configurations:

Using configuration file	Using environment variables
queue.bull.redis.username:USERNAME	QUEUE_BULL_REDIS_USERNAME
queue.bull.redis.password:PASSWORD	QUEUE_BULL_REDIS_PASSWORD
queue.bull.redis.db:0	QUEUE_BULL_REDIS_DB
queue.bull.redis.timeoutThreshold:10000ms	QUEUE_BULL_REDIS_TIMEOUT_

```
queue.bull.gracefulShutdownTimeout:30          N8N_GRACEFUL_SHUTDOWN_TIM
```



Now you can start your n8n instance and it will connect to your Redis instance.

Start workers

You will need to start worker processes to allow n8n to execute workflows. If you want to host workers on a separate machine, install n8n on the machine and make sure that it's connected to your Redis instance and the n8n database.

Start worker processes by running the following command from the root directory:

```
./packages/cli/bin/n8n worker
```

If you're using Docker, use the following command:

```
docker run --name n8n-queue -p 5679:5678 docker.n8n.io/n8nio/n8n
worker
```

You can set up multiple worker processes. Make sure that all the worker processes have access to Redis and the n8n database.

Worker server

Each worker process runs a server that exposes optional endpoints:

- `/healthz`: returns whether the worker is up, if you enable the `QUEUE_HEALTH_CHECK_ACTIVE` environment variable
- `/healthz/readiness`: returns whether worker's DB and Redis connections are ready, if you enable the `QUEUE_HEALTH_CHECK_ACTIVE` environment variable
- [credentials overwrite endpoint](#)
- [metrics](#)

View running workers

You can view running workers and their performance metrics in n8n by selecting **Settings > Workers**.

Running n8n with queues

When running n8n with queues, all the production workflow executions get processed by worker processes. This means that even the webhook calls get delegated to the worker processes, which might add some overhead and extra latency.

Redis acts as the message broker, and the database persists data, so access to both is required. Running a distributed system with this setup over SQLite isn't supported.

Webhook processors

Webhook processors are another layer of scaling in n8n. Configuring the webhook processor is optional, and allows you to scale the incoming webhook requests.

This method allows n8n to process a huge number of parallel requests. All you have to do is add more webhook processes and workers accordingly. The webhook process will listen to requests on the same port (default: 5678). Run these processes in containers or separate machines, and have a load balancing system to route requests accordingly.

n8n doesn't recommend adding the main process to the load balancer pool. If you add the main process to the pool, it will receive requests and possibly a heavy load. This will result in degraded performance for editing, viewing, and interacting with the n8n UI.

You can start the webhook processor by executing the following command from the root directory:

```
./packages/cli/bin/n8n webhook
```

If you're using Docker, use the following command:

```
docker run --name n8n-queue -p 5679:5678 -e "EXECUTIONS_MODE=queue"  
docker.n8n.io/n8nio/n8n webhook
```

Configure webhook URL

To configure your webhook URL, execute the following command on the machine running the main n8n instance:

```
export WEBHOOK_URL=https://your-webhook-url.com
```

You can also set this value in the configuration file.

Configure load balancer

When using multiple webhook processes you will need a load balancer to route requests. If you are using the same domain name for your n8n instance and the webhooks, you can set up your load balancer to route requests as follows:

- Redirect any request that matches /webhook/* to the webhook servers pool
- All other paths (the n8n internal API, the static files for the editor, etc.) should get routed to the main process

Note: The default URL for manual workflow executions is /webhook-test/*. Make sure that these URLs route to your main process.

You can change this path in the configuration file `endpoints.webhook` or using the `N8N_ENDPOINT_WEBHOOK` environment variable. If you change these, update your load balancer accordingly.

Disable webhook processing in the main process (optional)

You have webhook processors to execute the workflows. You can disable the webhook processing in the main process. This will make sure to execute all webhook executions in the webhook processors. In the configuration file set `endpoints.disableProductionWebhooksOnMainProcess` to `true` so that n8n doesn't process webhook requests on the main process.

Alternatively, you can use the following command:

```
export N8N_DISABLE_PRODUCTION_MAIN_PROCESS=true
```

When disabling the webhook process in the main process, run the main process and don't add it to the load balancer's webhook pool.

Configure worker concurrency

You can define the number of jobs a worker can run in parallel by using the `concurrency` flag. It defaults to 10. To change it:

```
n8n worker --concurrency=5
```

Concurrency and scaling recommendations

n8n recommends setting concurrency to 5 or higher for your worker instances. Setting low concurrency values with a large numbers of workers can exhaust your database's connection pool, leading to processing delays and failures.

Multi-main setup

In queue mode you can run more than one `main` process for high availability.

In a single-mode setup, the `main` process does two sets of tasks:

- **regular tasks**, such as running the API, serving the UI, and listening for webhooks, and
- **at-most-once tasks**, such as running non-HTTP triggers (timers, pollers, and persistent connections like RabbitMQ and IMAP), and pruning executions and binary data.

In a multi-main setup, there are two kinds of `main` processes:

- **followers**, which run **regular tasks**, and
- the **leader**, which runs **both regular and at-most-once tasks**.

Leader designation

In a multi-main setup, all main instances handle the leadership process transparently to users. In case the current leader becomes unavailable, for example because it crashed or its event loop became too busy, other followers can take over. If the previous leader becomes responsive again, it becomes a follower.

Configuring multi-main setup

To deploy n8n in multi-main setup, ensure:

- All `main` processes are running in queue mode and are connected to Postgres and Redis.
- All `main` and `worker` processes are running the same version of n8n.
- All `main` processes have set the environment variable `N8N_MULTI_MAIN_SETUP_ENABLED` to true.
- All `main` processes are running behind a load balancer with session persistence (sticky sessions) enabled.

If needed, you can adjust the leader key options:

Using configuration file	Using environment variables	D
<code>multiMainSetup.ttl:10</code>	<code>N8N_MULTI_MAIN_SETUP_KEY_TTL=10</code>	Ti (i) fo k m In se le in m
<code>multiMainSetup.interval:3</code>	<code>N8N_MULTI_MAIN_SETUP_CHECK_INTERVAL=3</code>	

Self-hosted concurrency control

In regular mode, n8n doesn't limit how many production executions may run at the same time. This can lead to a scenario where too many concurrent executions thrash the event loop, causing performance degradation and unresponsiveness.

To prevent this, you can set a concurrency limit for production executions in regular mode. Use this to control how many production executions run concurrently, and queue up any concurrent production executions over the limit. These executions remain in the queue until concurrency capacity frees up, and are then processed in FIFO order.

Concurrency control is disabled by default. To enable it:

```
export N8N_CONCURRENCY_PRODUCTION_LIMIT=20
```

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, error executions, or started from CLI.

- 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.
- To monitor concurrency control, watch logs for executions being added to the queue and released. In a future version, n8n will show concurrency control in the UI.

When you enable concurrency control, you can view the number of active executions and the configured limit at the top of a project's or workflow's executions tab.

Comparison to queue mode

In queue mode, you can control how many jobs a worker may run concurrently using the `--concurrency` flag.

Concurrency control in queue mode is a separate mechanism from concurrency control in regular mode, but the environment variable `N8N_CONCURRENCY_PRODUCTION_LIMIT` controls both of them. In queue mode, n8n takes the limit from this variable if set to a value other than -1, falling back to the `--concurrency` flag or its default.

Execution data

Depending on your executions settings and volume, your n8n database can grow in size and run out of storage.

To avoid this, n8n recommends that you don't save unnecessary data, and enable pruning of old executions data.

To do this, configure the corresponding [environment variables](#).

Reduce saved data

You can select which executions data n8n saves. For example, you can save only executions that result in an `Error`.

```
# npm
# Save executions ending in errors
export EXECUTIONS_DATA_SAVE_ON_ERROR=all

# Don't save successful executions
export EXECUTIONS_DATA_SAVE_ON_SUCCESS=none

# Don't save node progress for each execution
export EXECUTIONS_DATA_SAVE_ON_PROGRESS=false

# Don't save manually launched executions
export EXECUTIONS_DATA_SAVE_MANUAL_EXECUTIONS=false

# Docker
docker run -it --rm \
--name n8n \
```

```

-p 5678:5678 \
-e EXECUTIONS_DATA_SAVE_ON_ERROR=all \
-e EXECUTIONS_DATA_SAVE_ON_SUCCESS=none \
-e EXECUTIONS_DATA_SAVE_ON_PROGRESS=true \
-e EXECUTIONS_DATA_SAVE_MANUAL_EXECUTIONS=false \
docker.n8n.io/n8nio/n8n

# Docker Compose
n8n:
  environment:
    - EXECUTIONS_DATA_SAVE_ON_ERROR=all
    - EXECUTIONS_DATA_SAVE_ON_SUCCESS=none
    - EXECUTIONS_DATA_SAVE_ON_PROGRESS=true
    - EXECUTIONS_DATA_SAVE_MANUAL_EXECUTIONS=false

```

Enable executions pruning

Executions pruning deletes finished executions along with their execution data and binary data on a regular schedule. n8n enables pruning by default. For performance reasons, pruning first marks targets for deletion, and then later permanently removes them.

n8n prunes executions when **either** of the following condition occur:

- **Age:** The execution finished more than EXECUTIONS_DATA_MAX_AGE hours ago (default: 336 hours -> 14 days).
- **Count:** The total number of executions exceeds EXECUTIONS_DATA_PRUNE_MAX_COUNT (default: 10,000). When this occurs, n8n deletes executions from oldest to newest.

Keep in mind:

- Executions with the new, running, or waiting status aren't eligible for pruning.
- Annotated executions (for example, executions with tags or ratings) are never pruned.
- Pruning honors a safety buffer period of EXECUTIONS_DATA_HARD_DELETE_BUFFER hours (default: 1h), to ensure recent data remains available while the user is building or debugging a workflow.

```

# Enable executions pruning
export EXECUTIONS_DATA_PRUNE=true

# How old (hours) a finished execution must be to qualify for soft-deletion
export EXECUTIONS_DATA_MAX_AGE=168

# Max number of finished executions to keep. May not strictly prune back down to the exact max count. Set to `0` for unlimited.
export EXECUTIONS_DATA_PRUNE_MAX_COUNT=50000

# Docker
docker run -it --rm \
--name n8n \
-p 5678:5678 \
-e EXECUTIONS_DATA_PRUNE=true \
-e EXECUTIONS_DATA_MAX_AGE=168 \
docker.n8n.io/n8nio/n8n

```

```
# Docker Compose
n8n:
  environment:
    - EXECUTIONS_DATA_PRUNE=true
    - EXECUTIONS_DATA_MAX_AGE=168
    - EXECUTIONS_DATA_PRUNE_MAX_COUNT=50000
-8<- "_snippets/self-hosting/scaling/binary-data-pruning.md"
```

Binary data

Binary data is any file-type data, such as image files or documents generated or processed during the execution of a workflow.

Enable filesystem mode

When handling binary data, n8n keeps the data in memory by default. This can cause crashes when working with large files.

To avoid this, change the `N8N_DEFAULT_BINARY_DATA_MODE` [environment variable](#) to `filesystem`. This causes n8n to save data to disk, instead of using memory.

If you're using queue mode, switch this to database. n8n doesn't support `filesystem` mode with queue mode.

Binary data pruning

n8n executes binary data pruning as part of execution data pruning. Refer to [Execution data | Enable executions pruning](#) for details.

If you configure multiple binary data modes, binary data pruning operates on the active binary data mode. For example, if your instance stored data in S3, and you later switched to filesystem mode, n8n only prunes binary data in the filesystem. Refer to [External storage](#) for details.

External storage

n8n can store binary data produced by workflow executions externally. This feature is useful to avoid relying on the filesystem for storing large amounts of binary data.

n8n will introduce external storage for other data types in the future.

Storing n8n's binary data in S3

n8n supports [AWS S3](#) as an external store for binary data produced by workflow executions. You can use other S3-compatible services like Cloudflare R2 and Backblaze B2, but n8n doesn't officially support these.

Setup

Create and configure a bucket following the [AWS documentation](#). You can use the following policy, replacing <bucket-name> with the name of the bucket you created:

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "VisualEditor0",  
      "Effect": "Allow",  
      "Action": ["s3:*"],  
      "Resource": ["arn:aws:s3:::<bucket-name>", "arn:aws:s3:::<bucket-  
name>/*"]  
    }  
  ]  
}
```

Set a bucket-level lifecycle configuration so that S3 automatically deletes old binary data. n8n delegates pruning of binary data to S3, so setting a lifecycle configuration is required unless you want to preserve binary data indefinitely.

Once you finish creating the bucket, you will have a host, bucket name and region, and an access key ID and secret access key. You need to set them in n8n's environment:

```
export N8N_EXTERNAL_STORAGE_S3_HOST=... # example: s3.us-east-  
1.amazonaws.com  
export N8N_EXTERNAL_STORAGE_S3_BUCKET_NAME=...  
export N8N_EXTERNAL_STORAGE_S3_BUCKET_REGION=...  
export N8N_EXTERNAL_STORAGE_S3_ACCESS_KEY=...  
export N8N_EXTERNAL_STORAGE_S3_ACCESS_SECRET=...
```

Tell n8n to store binary data in S3:

```
export N8N_AVAILABLE_BINARY_DATA_MODES=filesystem,s3  
export N8N_DEFAULT_BINARY_DATA_MODE=s3
```

Restart the server to load the new configuration.

Usage

After you enable S3, n8n writes and reads any new binary data to and from the S3 bucket. n8n writes binary data to your S3 bucket in this format:

```
workflows/{workflowId}/executions/{executionId}/binary_data/{binaryF:  
|
```

n8n continues to read older binary data stored in the filesystem from the filesystem, if filesystem remains listed as an option in N8N_AVAILABLE_BINARY_DATA_MODES.

If you store binary data in S3 and later switch to filesystem mode, the instance continues to read any data stored in S3, as long as s3 remains listed in N8N_AVAILABLE_BINARY_DATA_MODES and your S3 credentials remain valid.

```
-8<- "_snippets/self-hosting/scaling/binary-data-pruning.md"
```

Memory-related errors

n8n doesn't restrict the amount of data each node can fetch and process. While this gives you freedom, it can lead to errors when workflow executions require more memory than available. This page explains how to identify and avoid these errors.

Identifying out of memory situations

n8n provides error messages that warn you in some out of memory situations. For example, messages such as **Execution stopped at this node (n8n may have run out of memory while executing it)**.

Error messages including **Problem running workflow**, **Connection Lost**, or **503 Service Temporarily Unavailable** suggest that an n8n instance has become unavailable.

When self-hosting n8n, you may also see error messages such as **Allocation failed - JavaScript heap out of memory** in your server logs.

On n8n Cloud, or when using n8n's Docker image, n8n restarts automatically when encountering such an issue. However, when running n8n with npm you might need to restart it manually.

Typical causes

Such problems occur when a workflow execution requires more memory than available to an n8n instance. Factors increasing the memory usage for a workflow execution include:

- Amount of JSON data.
- Size of binary data.
- Number of nodes in a workflow.
- Some nodes are memory-heavy: the Code node and the older Function node can increase memory consumption significantly.
- Manual or automatic workflow executions: manual executions increase memory consumption as n8n makes a copy of the data for the frontend.
- Additional workflows running at the same time.

Avoiding out of memory situations

When encountering an out of memory situation, there are two options: either increase the amount of memory available to n8n or reduce the memory consumption.

Increase available memory

When self-hosting n8n, increasing the amount of memory available to n8n means provisioning your n8n instance with more memory. This may incur additional costs with your hosting provider.

On n8n cloud you need to upgrade to a larger plan.

Reduce memory consumption

This approach is more complex and means re-building the workflows causing the issue. This section provides some guidelines on how to reduce memory consumption. Not all suggestions are applicable to all workflows.

-8<- “_snippets/self-hosting/scaling/reduce-memory-consumption.md”

Increase old memory

This applies to self-hosting n8n. When encountering **JavaScript heap out of memory** errors, it's often useful to allocate additional memory to the old memory section of the V8 JavaScript engine. To do this, set the appropriate V8 option `--max-old-space-size=SIZE` either through the CLI or through the NODE_OPTIONS environment variable.

Securing n8n

Securing your n8n instance can take several forms.

At a high level, you can:

- Conduct a security audit to identify security risks.
- Set up SSL to enforce secure connections.
- Set up Single Sign-On for user account management.
- Use two-factor authentication (2FA) for your users.

More granularly, consider blocking or opting out of features or data collection you don't want:

- Disable the public API if you aren't using it.
 - Opt out of data collection of the anonymous data n8n collects automatically.
 - Block certain nodes from being available to your users.
 - Restrict account registration to email-verified users.
-

Set up SSL

There are two methods to support TLS/SSL in n8n.

Use a reverse proxy (recommended)

Use a reverse proxy like Traefik or a Network Load Balancer (NLB) in front of the n8n instance. This should also take care of certificate renewals.

Refer to Security | Data encryption for more information.

Pass certificates into n8n directly

You can also choose to pass certificates into n8n directly. To do so, set the `N8N_SSL_CERT` and `N8N_SSL_KEY` environment variables to point to your generated certificate and key file.

You'll need to make sure the certificate stays renewed and up to date.

Refer to [Deployment environment variables](#) for more information on these variables and [Configuration](#) for more information on setting environment variables.

Set up Single Sign-On (SSO)

n8n supports the SAML and OIDC authentication protocols for single sign-on (SSO). See [OIDC vs SAML](#) for more general information on the two protocols, the differences between them, and their respective benefits.

- [Set up SAML](#): a general guide to setting up SAML in n8n, and links to resources for common identity providers (IdPs).
 - [Set up OIDC](#): a general guide to setting up OpenID Connect (OIDC) SSO in n8n.
-

Security audit

You can run a security audit on your n8n instance, to detect common security issues.

Run an audit

You can run an audit using the CLI, the public API, or the n8n node.

CLI

Run `n8n audit`.

API

Make a `POST` call to the `/audit` endpoint. You must authenticate as the instance owner.

n8n node

Add the [n8n node](#) to your workflow. Select **Resource > Audit** and **Operation > Generate**.

Report contents

The audit generates five risk reports:

Credentials

This report shows:

- Credentials not used in a workflow.
- Credentials not used in an active workflow.
- Credentials not used in a recently active workflow.

Database

This report shows:

- Expressions used in **Execute Query** fields in SQL nodes.
- Expressions used in **Query Parameters** fields in SQL nodes.
- Unused **Query Parameters** fields in SQL nodes.

File system

This report lists nodes that interact with the file system.

Nodes

This report shows:

- Official risky nodes. These are n8n built in nodes. You can use them to fetch and run any code on the host system, which exposes the instance to exploits. You can view the list in [n8n code | Audit constants](#), under OFFICIAL_RISKY_NODE_TYPES.
- Community nodes.
- Custom nodes.

Instance

This report shows:

- Unprotected webhooks in the instance.
- Missing security settings
- If your instance is outdated.

Disable the public REST API

The [n8n public REST API](#) allows you to programmatically perform many of the same tasks as you can in the n8n GUI.

If you don't plan on using this API, n8n recommends disabling it to improve the security of your n8n installation.

To disable the [public REST API](#), set the N8N_PUBLIC_API_DISABLED environment variable to true, for example:

```
export N8N_PUBLIC_API_DISABLED=true
```

Disable the API playground

To disable the [API playground](#), set the N8N_PUBLIC_API_SWAGGERUI_DISABLED environment variable to true, for example:

```
export N8N_PUBLIC_API_SWAGGERUI_DISABLED=true
```

Related resources

Refer to [Deployment environment variables](#) for more information on these environment variables.

Refer to [Configuration](#) for more information on setting environment variables.

Data collection

n8n collects some anonymous data from self-hosted n8n installations. Use the instructions below to opt out of data telemetry collection.

Collected data

Refer to [Privacy | Data collection in self-hosted n8n](#) for details on the data n8n collects.

How collection works

Your n8n instance sends most data to n8n as the events that generate it occur. Workflow execution counts and an instance pulse are sent periodically (every 6 hours). These data types mostly fall into n8n telemetry collection.

Opting out of data collection

n8n enables telemetry collection by default. To disable it, configure the following environment variables.

Opt out of telemetry events

To opt out of telemetry events, set the `N8N_DIAGNOSTICS_ENABLED` environment variable to false, for example:

```
export N8N_DIAGNOSTICS_ENABLED=false
```

Opt out of checking for new versions of n8n

To opt out of checking for new versions of n8n, set the `N8N_VERSION_NOTIFICATIONS_ENABLED` environment variable to false, for example:

```
export N8N_VERSION_NOTIFICATIONS_ENABLED=false
```

Disable all connection to n8n servers

If you want to fully prevent all communication with n8n's servers, refer to [Isolate n8n](#).

Related resources

Refer to [Deployment environment variables](#) for more information on these environment variables.

Refer to [Configuration](#) for more information on setting environment variables.

Block access to nodes

For security reasons, you may want to block your users from accessing or working with specific n8n nodes. This is helpful if your users might be untrustworthy.

Use the `NODES_EXCLUDE` environment variable to prevent your users from accessing specific nodes.

Exclude nodes

Update your `NODES_EXCLUDE` environment variable to include an array of strings containing any nodes you want to block your users from using.

For example, setting the variable this way:

```
NODES_EXCLUDE: ["\"n8n-nodes-base.executeCommand\"", \"n8n-nodes-base.readWriteFile\"]"]
```

Blocks the [Execute Command](#) and [Read/Write Files from Disk](#) nodes.

Your n8n users won't be able to search for or use these nodes.

Suggested nodes to block

The nodes that can pose security risks vary based on your use case and user profile. Here are some nodes you might want to start with:

- [Execute Command](#)
- [Read/Write Files from Disk](#)

Related resources

Refer to [Nodes environment variables](#) for more information on this environment variable.

Refer to [Configuration](#) for more information on setting environment variables.

Hardening task runners

Task runners are responsible for executing code from the Code node. While Code node executions are secure, you can follow these recommendations to further harden your task runners.

Run task runners as sidecars in external mode

To increase the isolation between the core n8n process and code in the Code node, run task runners in external mode. External task runners launch as separate containers, providing a fully isolated environment to execute the JavaScript defined in the Code node.

Restrict account registration to email-verified users

You can require all new accounts to be verified by email. This prevents malicious admins from registering accounts without email verification.

Prerequisites

- SMTP must be set up and n8n must be able to send emails.

How to restrict account registration

Set the environment variable `N8N_INVITE_LINKS_EMAIL_ONLY` to true. This locks down your instance so that only users with verified email addresses can register.

For more details on configuring SMTP, see [Set up SMTP](#).

Self-hosted AI Starter Kit

The Self-hosted AI Starter Kit is an open, docker compose template that bootstraps a fully featured Local AI and Low Code development environment.

Curated by [n8n](#), it combines the self-hosted n8n platform with a list of compatible AI products and components to get you started building self-hosted AI workflows.

What's included

- ✓ **Self-hosted n8n:** Low-code platform with over 400 integrations and advanced AI components.
- ✓ **Ollama:** Cross-platform LLM platform to install and run the latest local LLMs.

✓ **Odrant**: Open-source, high performance vector store with a comprehensive API.

✓ **PostgreSQL**: The workhorse of the Data Engineering world, handles large amounts of data safely.

What you can build

- ★ [AI Agents](#){ data-preview} that can schedule appointments
- ★ Summaries of company PDFs without leaking data
- ★ Smarter Slackbots for company communications and IT-ops
- ★ Private, low-cost analyses of financial documents

Get the kit

Head to [the GitHub repository](#) to clone the repo and get started!

Architecture

Understanding n8n's underlying architecture is helpful if you need to:

- Embed n8n
- Customize n8n's default databases

This section is a work in progress. If you have questions, please try the [forum](#) and let n8n know which architecture documents would be useful for you.

Database structure

This page describes the purpose of each table in the n8n database.

Database and query technology

By default, n8n uses SQLite as the database. If you are using another database the structure will be similar, but the data-types may be different depending on the database.

n8n uses [TypeORM](#) for queries and migrations.

To inspect the n8n database, you can use [DBeaver](#), which is an open-source universal database tool.

Tables

These are the tables n8n creates during setup.

auth_identity

Stores details of external authentication providers when using [SAML](#).

auth_provider_sync_history

Stores the history of a SAML connection.

credentials_entity

Stores the [credentials](#) used to authenticate with integrations.

event_destinations

Contains the destination configurations for [Log streaming](#).

execution_data

Contains the workflow at time of running, and the execution data.

execution_entity

Stores all saved workflow executions. Workflow settings can affect which executions n8n saves.

execution_metadata

Stores [Custom executions data](#).

installed_nodes

Lists the [community nodes](#) installed in your n8n instance.

installed_packages

Details of npm community nodes packages installed in your n8n instance. [installed_nodes](#) lists each individual node. [installed_packages](#) lists npm packages, which may contain more than one node.

migrations

A log of all database migrations. Read more about [Migrations](#) in TypeORM's documentation.

project

Lists the [projects](#) in your instance.

project_relation

Describes the relationship between a user and a [project](#), including the user's [role type](#).

role

Not currently used. For use in future work on custom roles.

settings

Records custom instance settings. These are settings that you can't control using environment variables. They include:

- Whether the instance owner is set up
- Whether the user chose to skip owner and user management setup
- Whether certain types of authentication, including SAML and LDAP, are on
- License key

shared_credentials

Maps credentials to users.

shared_workflow

Maps workflows to users.

tag_entity

All workflow tags created in the n8n instance. This table lists the tags. [workflows_tags](#) records which workflows have which tags.

user

Contains user data.

variables

Store [variables](#).

webhook_entity

Records the active webhooks in your n8n instance's workflows. This isn't just webhooks uses in the Webhook node. It includes all active webhooks used by any trigger node.

workflow_entity

Your n8n instance's saved workflows.

workflow_history

Store previous versions of workflows.

workflow_statistics

Counts workflow IDs and their status.

workflows_tags

Maps tags to workflows. `tag_entity` contains tag details.

Entity Relationship Diagram (ERD)

[Image: “n8n ERD”]

CLI commands for n8n

n8n includes a CLI (command line interface), allowing you to perform actions using the CLI rather than the n8n editor. These include starting workflows, and exporting and importing workflows and credentials.

Running CLI commands

You can use CLI commands with self-hosted n8n. Depending on how you choose to install n8n, there are differences in how to run the commands:

- npm: the `n8n` command is directly available. The documentation uses this in the examples below.
- Docker: the `n8n` command is available within your Docker container:

```
docker exec -u node -it <n8n-container-name> <n8n-cli-command>
```

Start a workflow

You can start workflows directly using the CLI.

Execute a saved workflow by its ID:

```
n8n execute --id <ID>
```

Change the active status of a workflow

You can change the active status of a workflow using the CLI.

Set the active status of a workflow by its ID to false:

```
n8n update:workflow --id=<ID> --active=false
```

Set the active status of a workflow by its ID to true:

```
n8n update:workflow --id=<ID> --active=true
```

Set the active status to false for all the workflows:

```
n8n update:workflow --all --active=false
```

Set the active status to true for all the workflows:

```
n8n update:workflow --all --active=true
```

Export entities

You can export your database entities from n8n using the CLI. This tooling allows you to export all entity types from one database type, such as SQLite, and import them into another database type, such as Postgres.

Command flags:

Flag	Description
-help	Help prompt.
-outputDir	Output directory path
-includeExecutionHistoryDataTables	Include execution history data tables, these are excluded by default as they can be very large

```
n8n export:entities --outputDir=./outputs --  
includeExecutionHistoryDataTables=true
```

Export workflows and credentials

You can export your workflows and credentials from n8n using the CLI.

Command flags:

Flag	Description
-help	Help prompt.
-all	Exports all workflows/credentials.
-backup	Sets -all -pretty -separate for backups. You can optionally set -output.
-id	The ID of the workflow to export.
-output	Outputs file name or directory if using separate files.
-pretty	Formats the output in an easier to read fashion.
-separate	Exports one file per workflow (useful for versioning). Must set a directory using -output.
-decrypted	Exports the credentials in a plain text format.

Workflows

Export all your workflows to the standard output (terminal):

```
n8n export:workflow --all
```

Export a workflow by its ID and specify the output file name:

```
n8n export:workflow --id=<ID> --output=file.json
```

Export all workflows to a specific directory in a single file:

```
n8n export:workflow --all --output=backups/latest/file.json
```

Export all the workflows to a specific directory using the --backup flag (details above):

```
n8n export:workflow --backup --output=backups/latest/
```

Credentials

Export all your credentials to the standard output (terminal):

```
n8n export:credentials --all
```

Export credentials by their ID and specify the output file name:

```
n8n export:credentials --id=<ID> --output=file.json
```

Export all credentials to a specific directory in a single file:

```
n8n export:credentials --all --output=backups/latest/file.json
```

Export all the credentials to a specific directory using the --backup flag (details above):

```
n8n export:credentials --backup --output=backups/latest/
```

Export all the credentials in plain text format. You can use this to migrate from one installation to another that has a different secret key in the configuration file.

```
n8n export:credentials --all --decrypted --  
output=backups/decrypted.json
```

Import entities

You can import entities from a previous export:entities command using this command, it allows importing of entities into a database type that differs from the exported database type. Current supported database types include: SQLite, Postgres.

The database is expected to be empty prior to import, this can be forced with the --truncateTables parameter.

Command flags:

Flag	Description
-help	Help prompt.
-inputDir	Input directory that holds output files for import

```
-truncateTables Truncate tables before import


---


n8n import:entities --inputDir ./outputs --truncateTables true
```

Import workflows and credentials

You can import your workflows and credentials from n8n using the CLI.

Available flags:

Flag	Description
-help	Help prompt.
-input	Input file name or directory if you use --separate.
-projectId	Import the workflow or credential to the specified project. Can't be used with --userId.
-separate	Imports *.json files from directory provided by -input.
-userId	Import the workflow or credential to the specified user. Can't be used with --projectId.

Workflows

Import workflows from a specific file:

```
n8n import:workflow --input=file.json
```

Import all the workflow files as JSON from the specified directory:

```
n8n import:workflow --separate --input=backups/latest/
```

Credentials

Import credentials from a specific file:

```
n8n import:credentials --input=file.json
```

Import all the credentials files as JSON from the specified directory:

```
n8n import:credentials --separate --input=backups/latest/
```

License

Clear

Clear your existing license from n8n's database and reset n8n to default features:

```
n8n license:clear
```

If your license includes [floating entitlements](#), running this command will also attempt to release them back to the pool, making them available for other instances.

Info

Display information about the existing license:

```
n8n license:info
```

User management

You can reset user management using the n8n CLI. This returns user management to its pre-setup state. It removes all user accounts.

Use this if you forget your password, and don't have SMTP set up to do password resets by email.

```
n8n user-management:reset
```

Disable MFA for a user

If a user loses their recovery codes you can disable MFA for a user with this command. The user will then be able to log back in to set up MFA again.

```
n8n mfa:disable --email=johndoe@example.com
```

Disable LDAP

You can reset the LDAP settings using the command below.

```
n8n ldap:reset
```

Uninstall community nodes and credentials

You can manage community nodes using the n8n CLI. For now, you can only uninstall community nodes and credentials, which is useful if a community node causes instability.

Command flags:

Flag	Description
-help	Show CLI help.
-credential	The credential type. Get this value by visiting the node's <NODE>.credential.ts file and getting the value of name.
-package	Package name of the community node.
-uninstall	Uninstalls the node.
-userId	The ID of the user who owns the credential. On self-hosted, query the database. On cloud, query the API with your API key.

Nodes

Uninstall a community node by package name:

```
n8n community-node --uninstall --package <COMMUNITY_NODE_NAME>
```

For example, to uninstall the [Evolution API community node](#), type:

```
n8n community-node --uninstall --package n8n-nodes-evolution-api
```

Credentials

Uninstall a community node credential:

```
n8n community-node --uninstall --credential <CREDENTIAL_TYPE> --  
userId <ID>
```

For example, to uninstall the [Evolution API community node credential](#), visit the [repository](#) and navigate to the [credentials.ts](#) file to find the name:

```
n8n community-node --uninstall --credential evolutionApi --userId  
1234
```

Security audit

You can run a [security audit](#) on your n8n instance, to detect common security issues.

```
n8n audit
```

Code in n8n

n8n is a low-code tool. This means you can do a lot without code, then add code when needed.

Code in your workflows

There are two places in your workflows where you can use code:

- **Expressions**

Use [expressions](#) to transform [data](#) in your nodes. You can use JavaScript in expressions, as well as n8n's [Built-in methods and variables](#) and [Data transformation functions](#).

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

- **Code node**

Use the Code node to add JavaScript or Python to your workflow.

[:octicons-arrow-right-24: Code node](#)

Other technical resources

These are features that are relevant to technical users.

Technical nodes

n8n provides core nodes, which simplify adding key functionality such as API requests, webhooks, scheduling, and file handling.

- **Write a backend**

The [HTTP Request](#), [Webhook](#), and [Code](#) nodes help you make API calls, respond to webhooks, and write any JavaScript in your workflow.

Use this do things like [Create an API endpoint](#).

[:octicons-arrow-right-24: Core nodes](#)

- **Represent complex logic**

You can build complex flows, using nodes like [If](#), [Switch](#), and [Merge](#) nodes.

[:octicons-arrow-right-24: Flow logic](#)

Other developer resources

- **The n8n API**

n8n provides an API, where you can programmatically perform many of the same tasks as you can in the GUI. There's an [n8n API node](#) to access the API in your workflows.

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

- **Self-host**

You can self-host n8n. This keeps your data on your own infrastructure.

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

- **Build your own nodes**

You can build custom nodes, install them on your n8n instance, and publish them to [npm](#).

[:octicons-arrow-right-24: Creating nodes](#)

Expressions

Expressions are a powerful feature implemented in all n8n nodes. They allow node parameters to be set dynamically based on data from:

- Previous node executions
- The workflow
- Your n8n environment

You can also execute JavaScript within an expression, making this a convenient and easy way to manipulate data into useful parameter values without writing extensive extra code.

n8n created and uses a templating language called [Tournament](#), and extends it with [custom methods and variables](#) and [data transformation functions](#). These features make it easier to perform common tasks like getting data from other nodes or accessing workflow metadata.

n8n additionally supports two libraries:

- [Luxon](#), for working with dates and time.
- [JMESPath](#), for querying JSON.

Writing expressions

To use an expression to set a parameter value:

1. Hover over the parameter where you want to use an expression.
2. Select **Expressions** in the **Fixed/Expression** toggle.
3. Write your expression in the parameter, or select **Open expression editor** Image: Open expressions editor icon to open the expressions editor. If you use the expressions editor, you can browse the available data in the **Variable selector**. All expressions have the format `{{ your expression here }}`.

Example: Get data from webhook body

Consider the following scenario: you have a webhook trigger that receives data through the webhook body. You want to extract some of that data for use in the workflow.

Your webhook data looks similar to this:

```
[  
 {  
   "headers": {  
     "host": "n8n.instance.address",  
     ...  
   },  
   "params": {},  
   "query": {},  
   "body": {  
     "name": "Jim",  
     "age": 30,  
     "city": "New York"  
   }  
 }
```

In the next node in the workflow, you want to get just the value of city. You can use the following expression:

```
 {{$json.body.city}}
```

This expression:

1. Accesses the incoming JSON-formatted data using n8n's custom `$json` variable.
2. Finds the value of city (in this example, "New York"). Note that this example uses JMESPath syntax to query the JSON data. You can also write this expression as `>{{$json['body']['city']}}`.

Example: Writing longer JavaScript

You can do things like variable assignments or multiple statements in an expression, but you need to wrap your code using the syntax for an IIFE (Immediately Invoked Function Expression).

The following code use the Luxon date and time library to find the time between two dates in months. We surround the code in both the handlebar brackets for an expression and the IIFE syntax.

```
{{{()=>{
  let end = DateTime.fromISO('2017-03-13');
  let start = DateTime.fromISO('2017-02-13');
  let diffInMonths = end.diff(start, 'months');
  return diffInMonths.toObject();
}}()}}
```

Common issues

For common errors or issues with expressions and suggested resolution steps, refer to [Common Issues](#).

Using the Code node

-8<- “_snippets/integrations/builtin/core-nodes/code-node.md”

AI coding with GPT

Not available on self-hosted.

Python isn't supported. ///

Use AI in the Code node

-8<- “_snippets/code/ai-how-to.md”

Usage limits

During the trial phase there are no usage limits. If n8n makes the feature permanent, there may be usage limits as part of your pricing tier.

Feature limits

The ChatGPT implementation in n8n has the following limitations:

- The AI writes code that manipulates data from the n8n workflow. You can't ask it to pull in data from other sources.
- The AI doesn't know your data, just the schema, so you need to tell it things like how to find the data you want to extract, or how to

check for null.

- Nodes before the Code node must execute and deliver data to the Code node before you run your AI query.
- Doesn't work with large incoming data schemas.
- May have issues if there are a lot of nodes before the code node.

Writing good prompts

Writing good prompts increases the chance of getting useful code back.

Some general tips:

- Provide examples: if possible, give a sample expected output. This helps the AI to better understand the transformation or logic you're aiming for.
- Describe the processing steps: if there are specific processing steps or logic that should apply to the data, list them in sequence. For example: "First, filter out all users under 18. Then, sort the remaining users by their last name."
- Avoid ambiguities: while the AI understands various instructions, being clear and direct ensures you get the most accurate code. Instead of saying "Get the older users," you might say "Filter users who are 60 years and above."
- Be clear about what you expect as the output. Do you want the data transformed, filtered, aggregated, or sorted? Provide as much detail as possible.

And some n8n-specific guidance:

- Think about the input data: make sure ChatGPT knows which pieces of the data you want to access, and what the incoming data represents. You may need to tell ChatGPT about the availability of n8n's built-in methods and variables.
- Declare interactions between nodes: if your logic involves data from multiple nodes, specify how they should interact. "Merge the output of 'Node A' with 'Node B' based on the 'userID' property". if you prefer data to come from certain nodes or to ignore others, be clear: "Only consider data from the 'Purchases' node and ignore the 'Refunds' node."
- Ensure the output is compatible with n8n. Refer to [Data structure](#) for more information on the data structure n8n requires.

Example prompts

These examples show a range of possible prompts and tasks.

Example 1: Find a piece of data inside a second dataset

To try the example yourself, [download the example workflow](#) and import it into n8n.

In the third Code node, enter this prompt:

The slack data contains only one item. The input data represents all Notion users. Sometimes the person property that holds the email can be null. I want to find the notionId of the Slack user and return it.

Take a look at the code the AI generates.

This is the JavaScript you need:

```
const slackUser = $("Mock Slack").all()[0];
const notionUsers = $input.all();
const slackUserEmail = slackUser.json.email;

const notionUser = notionUsers.find(
  (user) => user.json.person && user.json.person.email ===
slackUserEmail
);

return notionUser ? [{ json: { notionId: notionUser.json.id } }] : [];
```

Example 2: Data transformation

To try the example yourself, [download the example workflow](#) and import it into n8n.

In the **Join items** Code node, enter this prompt:

Return a single line of text that has all usernames listed with a comma. Each username should be enquoted with a double quotation mark.

Take a look at the code the AI generates.

This is the JavaScript you need:

```
const items = $input.all();
const usernames = items.map((item) => `"${item.json.username}"`);
const result = usernames.join(", ");
return [{ json: { usernames: result } }];
```

Example 3: Summarize data and create a Slack message

To try the example yourself, [download the example workflow](#) and import it into n8n.

In the **Summarize** Code node, enter this prompt:

Create a markdown text for Slack that counts how many ideas, features and bugs have been submitted. The type of submission is saved in the property_type field. A feature has the property “Feature”, a bug has the property “Bug” and an idea has the property “Bug”. Also, list the five top submissions by vote in that message. Use “<url|text>” as markdown for links.

Take a look at the code the AI generates.

This is the JavaScript you need:

```
const submissions = $input.all();

// Count the number of ideas, features, and bugs
let ideaCount = 0;
let featureCount = 0;
let bugCount = 0;
```

```

    submissions.forEach((submission) => {
      switch (submission.json.property_type[0]) {
        case "Idea":
          ideaCount++;
          break;
        case "Feature":
          featureCount++;
          break;
        case "Bug":
          bugCount++;
          break;
      }
    });

    // Sort submissions by votes and take the top 5
    const topSubmissions = submissions
      .sort((a, b) => b.json.property_votes - a.json.property_votes)
      .slice(0, 5);

    let topSubmissionText = "";
    topSubmissions.forEach((submission) => {
      topSubmissionText +=
`<${submission.json.url}|${submission.json.name}> with
${submission.json.property_votes} votes\n`;
    });

    // Construct the Slack message
    const slackMessage = `*Summary of Submissions*\n
Ideas: ${ideaCount}\n
Features: ${featureCount}\n
Bugs: ${bugCount}\n
Top 5 Submissions:\n
${topSubmissionText}`;

    return [{ json: { slackMessage } }];
  }
}

```

Reference incoming node data explicitly

If your incoming data contains nested fields, using dot notation to reference them can help the AI understand what data you want.

[Image: “Screenshot of an n8n code node, highlighting how to reference data with dot notation in an AI query”]

To try the example yourself, [download the example workflow](#) and import it into n8n.

In the second Code node, enter this prompt:

The data in “Mock data” represents a list of people. For each person, return a new item containing personal_info.first_name and work_info.job_title.

This is the JavaScript you need:

```

const items = $input.all();
const newItems = items.map((item) => {
  const firstName = item.json.personal_info.first_name;
  const jobTitle = item.json.work_info.job_title;
  return {
    json: {

```

```
        firstName,  
        jobTitle,  
    },  
};  
});  
return newItems;
```

Related resources

Pluralsight offer a short guide on [How to use ChatGPT to write code](#), which includes example prompts.

Fixing the code

The AI-generated code may work without any changes, but you may have to edit it. You need to be aware of n8n's [Data structure](#). You may also find n8n's built-in methods and variables useful.

Built-in methods and variables

n8n provides built-in methods and variables for working with data and accessing n8n data. This section provides a reference of available methods and variables for use in [expressions](#), with a short description.

The [Cookbook](#) contains examples for some common tasks, including some [Code node only](#) functions.

Current node input

Methods for working with the input of the current node. Some methods and variables aren't available in the Code node.

```
==== "JavaScript" | Method | Description | Available in Code node? ||  
--- | --- | :-----: | | $binary | Shorthand for  
$input.item.binary. Incoming binary data from a node | :x: ||  
$input.item | The input item of the current node that's being  
processed. Refer to Item linking for more information on paired items  
and item linking. | :white_check_mark: || $input.all() | All input items  
in current node. | :white_check_mark: || $input.first() | First input  
item in current node. | :white_check_mark: || $input.last() | Last  
input item in current node. | :white_check_mark: || $input.params |  
Object containing the query settings of the previous node. This  
includes data such as the operation it ran, result limits, and so on. |  
:white_check_mark: || $json | Shorthand for $input.item.json.  
Incoming JSON data from a node. Refer to Data structure for  
information on item structure. | :white_check_mark: (when running  
once for each item) || $input.context.noItemsLeft | Boolean. Only  
available when working with the Loop Over Items node. Provides  
information about what's happening in the node. Use this to  
determine whether the node is still processing items. |  
:white_check_mark: | === "Python" | Method | Description | | --- |  
--- | | _input.item | The input item of the current node that's being  
processed. Refer to Item linking for more information on paired items
```

and item linking. | | `_input.all()` | All input items in current node. | | `_input.first()` | First input item in current node. | | `_input.last()` | Last input item in current node. | | `_input.params` | Object containing the query settings of the previous node. This includes data such as the operation it ran, result limits, and so on. | | `_json` | Shorthand for `_input.item.json`. Incoming JSON data from a node. Refer to [Data structure](#) for information on item structure. Available when you set **Mode to Run Once for Each Item**. | | `_input.context.noItemsLeft` | Boolean. Only available when working with the Loop Over Items node. Provides information about what's happening in the node. Use this to determine whether the node is still processing items. |

Output of other nodes

Methods for working with the output of other nodes. Some methods and variables aren't available in the Code node.

```
==== "JavaScript" | Method | Description | Available in Code node? | |
--- | --- | :-----: | | $("<node-name>").all(branchIndex?, runIndex?) | Returns all items from a given node. If branchIndex isn't given it will default to the output that connects node-name with the node where you use the expression or code. | :white_check_mark: | | $("<node-name>").first(branchIndex?, runIndex?) | The first item output by the given node. If branchIndex isn't given it will default to the output that connects node-name with the node where you use the expression or code. | :white_check_mark: | | $("<node-name>").last(branchIndex?, runIndex?) | The last item output by the given node. If branchIndex isn't given it will default to the output that connects node-name with the node where you use the expression or code. | :white_check_mark: | | $("<node-name>").item | The linked item. This is the item in the specified node used to produce the current item. Refer to Item linking for more information on item linking. | :white_check_mark: | | $("<node-name>").params | Object containing the query settings of the given node. This includes data such as the operation it ran, result limits, and so on. | :white_check_mark: | | $("<node-name>").context | Boolean. Only available when working with the Loop Over Items node. Provides information about what's happening in the node. Use this to determine whether the node is still processing items. | :white_check_mark: | | $("<node-name>").itemMatching(currentNodeInputIndex) | Use instead of $("<node-name>").item in the Code node if you need to trace back from an input item. | :white_check_mark: | === "Python" | Method | Description | Available in Code node? | | --- | --- | :-----: | | ("<node-name>").all(branchIndex?, runIndex?) | Returns all items from a given node. If branchIndex isn't given it will default to the output that connects node-name with the node where you use the expression or code. | :white_check_mark: | | ("<node-name>").first(branchIndex?, runIndex?) | The first item output by the given node. If branchIndex isn't given it will default to the output that connects node-name with the node where you use the expression or code. | :white_check_mark: | | ("<node-name>").last(branchIndex?, runIndex?) | The last item output by the given node. If branchIndex isn't given it will default to the output that connects node-name with the node where you use the expression or code. | :white_check_mark: | | ("<node-name>").item | The linked item. This is the item in the specified node used to produce the current item. Refer to Item linking for more information on item linking. | :white_check_mark: | | ("<node-name>").params | Object containing the query settings of the given node. This includes data
```

such as the operation it ran, result limits, and so on. |
:white_check_mark: || _("<node-name>").context | Boolean. Only available when working with the Loop Over Items node. Provides information about what's happening in the node. Use this to determine whether the node is still processing items. |
:white_check_mark: || _("<node-name>").itemMatching(currentNodeInputIndex) | Use instead of _("<node-name>").item in the Code node if you need to trace back from an input item. Refer to [Retrieve linked items from earlier in the workflow](#) for an example. | :white_check_mark: |

Built-in date and time methods

Methods for working with date and time.

==== "JavaScript" | Method | Description | Available in Code node? ||
--- | ---- | :-----: | \$now | A Luxon object containing the current timestamp. Equivalent to DateTime.now(). |
:white_check_mark: || \$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 }). | :white_check_mark: | == "Python" | Method | Description ||
--- | ---- | | _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 }). |

n8n provides built-in convenience functions to support data transformation in expressions for dates. Refer to [Data transformation functions | Dates](#) for more information.

JMESPath method

This is an n8n-provided method for working with the [JMESPath library](#).

==== "JavaScript" | Method | Description | Available in Code node? ||
--- | ---- | :-----: | | \$jmespath() | Perform a search on a JSON object using JMESPath. | :white_check_mark: | == "Python" | Method | Description || --- | ---- | | _jmespath() | Perform a search on a JSON object using JMESPath. |

HTTP node variables

Variables for working with HTTP node requests and responses when using pagination.

Refer to [HTTP Request](#) for guidance on using the HTTP node, including configuring pagination.

Refer to [HTTP Request node cookbook | Pagination](#) for example pagination configurations.

-8<- “ snippets/integrations/builtin/core-nodes/http/pagination-variables.md”

LangChain Code node methods

n8n provides these methods to make it easier to perform common tasks in the [LangChain Code node](#).

-8<- “ snippets/integrations/builtin/cluster-nodes/langchain-root-nodes/langchaincode/builtin-methods.md”

n8n metadata

Methods for working with n8n metadata.

This includes:

- Access to n8n environment variables for self-hosted n8n.
- Metadata about workflows, executions, and nodes.
- Information about instance [Variables](#) and [External secrets](#).

```
==== "JavaScript" | Method | Description | Available in Code node? ||  
---- | ----- | :-----: | $env | Contains n8n instance  
configuration environment variables. | :white_check_mark: ||  
$execution.customData | Set and get custom execution data. Refer to  
Custom executions data for more information. | :white_check_mark: ||  
$execution.id | The unique ID of the current workflow execution. |  
:white_check_mark: || $execution.mode | Whether the execution was  
triggered automatically, or by manually running the workflow. Possible  
values are test and production. | :white_check_mark: ||  
$execution.resumeUrl | The webhook URL to call to resume a workflow  
waiting at a Wait node. | :white_check_mark: ||  
$getWorkflowStaticData(type) | View an example. Static data doesn't  
persist when testing workflows. The workflow must be active and  
called by a trigger or webhook to save static data. This gives access to  
the static workflow data. | :white_check_mark: || $(<node-  
name>).isExecuted | Check whether a node has already executed. |  
:white_check_mark: || $itemIndex | The index of an item in a list of  
items. | :x: || $nodeVersion | Get the version of the current node. |  
:white_check_mark: || $prevNode.name | The name of the node that the  
current input came from. When using the Merge node, note that  
$prevNode always uses the first input connector. | :white_check_mark: |  
| $prevNode.outputIndex | The index of the output connector that the  
current input came from. Use this when the previous node had  
multiple outputs (such as an If or Switch node). When using the  
Merge node, note that $prevNode always uses the first input connector.  
| :white_check_mark: || $prevNode.runIndex | The run of the previous  
node that generated the current input. When using the Merge node,  
note that $prevNode always uses the first input connector.  
| :white_check_mark: || $runIndex | How many times n8n has executed  
the current node. Zero-based (the first run is 0, the second is 1, and so  
on). | :white_check_mark: || $secrets | Contains information about  
your External secrets setup. | :x: || $vars | Contains the Variables  
available in the active environment. | :white_check_mark: || $version |  
The node version. | :x: || $workflow.active | Whether the workflow is
```

active (true) or not (false). | :white_check_mark: || \$workflow.id | The workflow ID. | :white_check_mark: || \$workflow.name | The workflow name. | :white_check_mark: | === "Python (native)" | Method | Description || --- | ----- || _items | Contains incoming items in "Run once for all items" mode. || _item | Contains the item being iterated on in "Run once for each item" mode. | === "Python (Pyodide, deprecated)" | Method | Description || --- | ----- || _env | Contains n8n instance configuration environment variables. || _execution.customData | Set and get custom execution data. Refer to Custom executions data for more information. || _execution.id | The unique ID of the current workflow execution. || _execution.mode | Whether the execution was triggered automatically, or by manually running the workflow. Possible values are test and production. || _execution.resumeUrl | The webhook URL to call to resume a workflow waiting at a Wait node. || _getWorkflowStaticData(type) | View an example. Static data doesn't persist when testing workflows. The workflow must be active and called by a trigger or webhook to save static data. This gives access to the static workflow data. || ("<node-name>").isExecuted | Check whether a node has already executed. || _nodeVersion | Get the version of the current node. | :white_check_mark: || _prevNode.name | The name of the node that the current input came from. When using the Merge node, note that _prevNode always uses the first input connector. || _prevNode.outputIndex | The index of the output connector that the current input came from. Use this when the previous node had multiple outputs (such as an If or Switch node). When using the Merge node, note that _prevNode always uses the first input connector. || _prevNode.runIndex | The run of the previous node that generated the current input. When using the Merge node, note that _prevNode always uses the first input connector. || _runIndex | How many times n8n has executed the current node. Zero-based (the first run is 0, the second is 1, and so on). || _secrets | Contains information about your External secrets setup. || _vars | Contains the Variables available in the active environment. || _workflow.active | Whether the workflow is active (true) or not (false). || _workflow.id | The workflow ID. || _workflow.name | The workflow name. |

Convenience methods

n8n provides these methods to make it easier to perform common tasks in expressions.

==== "JavaScript" | Method | Description | Available in Code node? || --- | ----- | :-----: | \$evaluateExpression(expression: string, itemIndex?: number) | Evaluates a string as an expression. If you don't provide itemIndex, n8n uses the data from item 0 in the Code node. | :white_check_mark: || \$isEmpty(value, defaultValue) | The \$isEmpty() function takes two parameters, tests the first to check if it's empty, then returns either the parameter (if not empty) or the second parameter (if the first is empty). The first parameter is empty if it's:

- undefined
- null
- An empty string ''
- An array where value.length returns false
- An object where Object.keys(value).length returns false

| :white_check_mark: || \$if() | The \$if() function takes three

parameters: a condition, the value to return if true, and the value to return if false. | :x: || \$max() | Returns the highest of the provided numbers. | :x: || \$min() | Returns the lowest of the provided numbers. | :x: | === "Python" | Method | Description | — | — | | _evaluateExpression(expression: string, itemIndex?: number) | Evaluates a string as an expression. If you don't provide itemIndex, n8n uses the data from item 0 in the Code node. | | _isEmpty(value, defaultValue) | The _isEmpty() function takes two parameters, tests the first to check if it's empty, then returns either the parameter (if not empty) or the second parameter (if the first is empty). The first parameter is empty if it's:

- undefined
- null
- An empty string ''
- An array where value.length returns false
- An object where Object.keys(value).length returns false

| :white_check_mark: |

Data transformation functions

Data transformation functions are helper functions to make data transformation easier in expressions.

For a list of available functions, refer to the page for your data type:

- [Arrays](#)
- [Dates](#)
- [Numbers](#)
- [Objects](#)
- [Strings](#)

Usage

Data transformation functions are available in the expressions editor.

The syntax is:

```
{{ dataItem.function() }}
```

For example, to check if a string is an email:

```
{{ "example@example.com".isEmail() }}
```

// Returns true

Arrays

A reference document listing built-in convenience functions to support data transformation in expressions for arrays.

Booleans

A reference document listing built-in convenience functions to support data transformation in [expressions](#) for arrays.

Dates

A reference document listing built-in convenience functions to support data transformation in [expressions](#) for dates.

Numbers

A reference document listing built-in convenience functions to support data transformation in [expressions](#) for numbers.

Objects

A reference document listing built-in convenience functions to support data transformation in [expressions](#) for objects.

Strings

A reference document listing built-in convenience functions to support data transformation in [expressions](#) for strings.

Custom variables

Custom variables are read-only variables that you can use to store and reuse values in n8n workflows.

Create variables

You can access the **Variables** tab from either the overview page or a specific project.

To create a new variable:

1. On the **Variables** tab, select **Add Variable**.
2. Enter a **Key** and **Value**. The maximum key length is 50 characters, and the maximum value length is 1000 characters. n8n limits the characters you can use in the key and value to lowercase and uppercase letters, numbers, and underscores (A-Z, a-z, 0-9, _).
3. Select the **Scope** (only available when creating from the overview page):

- **Global:** The variable is available across all projects in the n8n instance.
 - **Project:** The variable is available only within a specific project (you can select which project).
 - When creating from a project page, the scope is automatically set to that project.
4. Select **Save**. The variable is now available for use in workflows according to its scope.

Edit and delete variables

To edit or delete a variable:

1. On the **Variables** tab, hover over the variable you want to change.
2. Select **Edit** or **Delete**.

Use variables in workflows

You can access variables in the Code node and in expressions:

```
// Access a variable  
$vars.<variable-name>
```

All variables are strings.

During workflow execution, n8n replaces the variables with the variable value. If the variable has no value, n8n treats its value as `undefined`. Workflows don't automatically fail in this case.

Variables are read-only. You must use the UI to change the values. If you need to set and access custom data within your workflow, use Workflow static data.

Date and time with Luxon

Luxon is a JavaScript library that makes it easier to work with date and time. For full details of how to use Luxon, refer to [Luxon's documentation](#).

n8n passes dates between nodes as strings, so you need to parse them. Luxon makes this easier.

Date and time behavior in n8n

Be aware of the following:

- In a workflow, n8n converts dates and times to strings between nodes. Keep this in mind when doing arithmetic on dates and times from other nodes.
- With vanilla JavaScript, you can convert a string to a date with `new Date('2019-06-23')`. In Luxon, you must use a function explicitly stating the format, such as `DateTime.fromISO('2019-06-23')` or `DateTime.fromFormat("23-06-2019", "dd-MM-yyyy")`.

Setting the timezone in n8n

Luxon uses the n8n timezone. This value is either:

- Default: America/New_York
- A custom timezone for your n8n instance, set using the `GENERIC_TIMEZONE` environment variable.
- A custom timezone for an individual workflow, configured in workflow settings.

Common tasks

This section provides examples for some common operations. More examples, and detailed guidance, are available in [Luxon's own documentation](#).

Get the current datetime or date

Use the `$now` and `$today` Luxon objects to get the current time or day:

- `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 })`.

Note that these variables can return different time formats when cast as a string:

```
==== "Expressions (JavaScript)" javascript      {{$now}}      // n8n
displays the ISO formatted timestamp      // For example 2022-03-
09T14:02:37.065+00:00      {"Today's date is " + $now}      // n8n
displays "Today's date is <unix timestamp>"      // For example
"Today's date is 1646834498755"

==== "Code node (JavaScript)" javascript      $now      // n8n
displays <ISO formatted timestamp>      // For example 2022-03-
09T14:00:25.058+00:00      let rightNow = "Today's date is " + $now
// n8n displays "Today's date is <unix timestamp>"      // For
example "Today's date is 1646834498755" ====
"Code node (Python)" python      _now      # n8n displays <ISO formatted timestamp>      #
For example 2022-03-09T14:00:25.058+00:00      rightNow = "Today's
date is " + str(_now)      # n8n displays "Today's date is <unix
timestamp>"      # For example "Today's date is 1646834498755"
```

n8n provides built-in convenience functions to support data transformation in expressions for dates. Refer to [Data transformation functions | Dates](#) for more information.

Convert JavaScript dates to Luxon

To convert a native JavaScript date to a Luxon date:

- In expressions, use the `.toDateTime()` method. For example, `{} (new Date()).toDateTime() {}`.
- In the Code node, use `DateTime.fromJSDate()`. For example, let `luxondate = DateTime.fromJSDate(new Date())`.

Convert date string to Luxon

You can convert date strings and other date formats to a Luxon DateTime object. You can convert from standard formats and from arbitrary strings.

If you have a date in a supported standard technical format:

Most dates use `fromISO()`. This creates a Luxon DateTime from an ISO 8601 string. For example:

```
==== "Expressions (JavaScript)"  
```js  
{ {DateTime.fromISO('2019-06-23T00:00:00.00') } }
````  
==== "Code node (JavaScript)"  
```js  
let luxonDateTime = DateTime.fromISO('2019-06-23T00:00:00.00')
```
```

Luxon's API documentation has more information on [fromISO](#).

Luxon provides functions to handle conversions for a range of formats. Refer to Luxon's guide to [Parsing technical formats](#) for details.

If you have a date as a string that doesn't use a standard format:

Use Luxon's [Ad-hoc parsing](#). To do this, use the `fromFormat()` function, providing the string and a set of `tokens` that describe the format.

For example, you have n8n's founding date, 23rd June 2019, formatted as 23-06-2019. You want to turn this into a Luxon object:

```
==== "Expressions (JavaScript)"  
```js  
{ {DateTime.fromFormat("23-06-2019", "dd-MM-yyyy") } }
````  
==== "Code node (JavaScript)"  
```js  
let newFormat = DateTime.fromFormat("23-06-2019", "dd-MM-yyyy")
```
```

When using ad-hoc parsing, note Luxon's warning about [Limitations](#). If you see unexpected results, try their [Debugging](#) guide.

Get n days from today

Get a number of days before or after today.

```
==== "Expressions (JavaScript)"
```

For example, you want to set a field to always show the date seven days before the current date.

In the expressions editor, enter:

```
```js
{{$today.minus({days: 7})}}
```
```

On the 23rd June 2019, this returns `"[Object: "2019-06-16T00:00:00.000+00:00"]`.

This example uses n8n's custom variable `\$today` for convenience. It's the equivalent of `DateTime.now().set({ hour: 0, minute: 0, second: 0, millisecond: 0 }).minus({days: 7})`.

==== “Code node (JavaScript)”

For example, you want a variable containing the date seven days before the current date.

In the code editor, enter:

```
```js
let sevenDaysAgo = $today.minus({days: 7})
```
```

On the 23rd June 2019, this returns `"[Object: "2019-06-16T00:00:00.000+00:00"]`.

This example uses n8n's custom variable `\$today` for convenience. It's the equivalent of `DateTime.now().set({ hour: 0, minute: 0, second: 0, millisecond: 0 }).minus({days: 7})`.

For more detailed information and examples, refer to:

- Luxon's [guide to math](#)
- Their API documentation on [DateTime plus](#) and [DateTime minus](#)

Create human-readable dates

In [Get n days from today](#), the example gets the date seven days before the current date, and returns it as `[Object: "yyyy-mm-dd-T00:00:00.000+00:00"]` (for expressions) or `yyyy-mm-dd-T00:00:00.000+00:00` (in the Code node). To make this more readable, you can use Luxon's formatting functions.

For example, you want the field containing the date to be formatted as DD/MM/YYYY, so that on the 23rd June 2019, it returns 23/06/2019.

This expression gets the date seven days before today, and converts it to the DD/MM/YYYY format.

==== “Expressions (JavaScript)”

```
```js
{{$today.minus({days: 7}).toLocaleString()}}
```
```

==== “Code node (JavaScript)”

```
```js
let readableSevenDaysAgo = $today.minus({days: 7}).toLocaleString()
```
```

```
```
```

You can alter the format. For example:

```
==== "Expressions (JavaScript)"
```

```
```js
{{$today.minus({days: 7}).toLocaleString({month: 'long', day:
'numeric', year: 'numeric'})}}
```
```

On 23rd June 2019, this returns "16 June 2019".

```
==== "Code node (JavaScript)"
```

```
```js
let readableSevenDaysAgo = $today.minus({days:
7}).toLocaleString({month: 'long', day: 'numeric', year: 'numeric'})
```
```

On 23rd June 2019, this returns "16 June 2019".

Refer to Luxon's guide on [toLocaleString \(strings for humans\)](#) for more information.

## Get the time between two dates

To get the time between two dates, use Luxon's `diffs` feature. This subtracts one date from another and returns a duration.

For example, get the number of months between two dates:

```
==== "Expressions (JavaScript)"
```

```
```js
{$Date.diff($Date.fromISO('2019-06-23'), $Date.fromISO('2019-05-
23'), 'months').toObject()}
```
```

This returns `'[Object: {"months":1}]`.

```
==== "Code node (JavaScript)"
```

```
```js
let monthsBetweenDates = $Date.diff($Date.fromISO('2019-06-
23'), $Date.fromISO('2019-05-23'), 'months').toObject()
```
```

This returns `{"months":1}`.

Refer to Luxon's [Diffs](#) for more information.

## A longer example: How many days to Christmas?

This example brings together several Luxon features, uses JMESPath, and does some basic string manipulation.

The scenario: you want a countdown to 25th December. Every day, it should tell you the number of days remaining to Christmas. You don't want to update it for next year - it needs to seamlessly work for every year.

```
==== "Expressions (JavaScript)"
```

```
```js
{{"There are " + $today.diff(DateTime.fromISO($today.year + '-12-25'), 'days').toObject().days.toString().substring(1) + " days to Christmas!"}}
```
```

This outputs `"There are <number of days> days to Christmas!"`. For example, on 9th March, it outputs "There are 291 days to Christmas!".

A detailed explanation of what the expression does:

- \* `{{` indicates the start of the expression.
- \* `"There are `: a string.
- \* `+`: used to join two strings.
- \* `'\$today.diff()`: This is similar to the example in [Get the time between two dates](#get-the-time-between-two-dates), but it uses n8n's custom `\$today` variable.
- \* `DateTime.fromISO(\$today.year + '-12-25'), 'days'`: this part gets the current year using `\$today.year`, turns it into an ISO string along with the month and date, and then takes the whole ISO string and converts it to a Luxon DateTime data structure. It also tells Luxon that you want the duration in days.
- \* `ToObject()` turns the result of diff() into a more usable object. At this point, the expression returns `[Object: {"days": -<number-of-days>} ]`. For example, on 9th March, `[Object: {"days": -291}]`.
- \* `days` uses JMESPath syntax to retrieve just the number of days from the object. For more information on using JMESPath with n8n, refer to our [JMESPpath](/code/cookbook/jmespath.md) documentation. This gives you the number of days to Christmas, as a negative number.
- \* `toString().substring(1)` turns the number into a string and removes the `-`.
- \* `+ " days to Christmas!"`: another string, with a `+` to join it to the previous string.
- \* `}}`: indicates the end of the expression.

```
==== "Code node (JavaScript)"
```

```
```js
let daysToChristmas = "There are " +
$today.diff(DateTime.fromISO($today.year + '-12-25'),
'days').toObject().days.toString().substring(1) + " days to
Christmas!";
```
```

This outputs `"There are <number of days> days to Christmas!"`. For example, on 9th March, it outputs "There are 291 days to Christmas!".

A detailed explanation of what the code does:

- \* `"There are `: a string.
- \* `+`: used to join two strings.
- \* `'\$today.diff()`: This is similar to the example in [Get the time between two dates](#get-the-time-between-two-dates), but it uses n8n's custom `\$today` variable.
- \* `DateTime.fromISO(\$today.year + '-12-25'), 'days'`: this part gets the current year using `\$today.year`, turns it into an ISO string

along with the month and date, and then takes the whole ISO string and converts it to a Luxon DateTime data structure. It also tells Luxon that you want the duration in days.

- \* `toObject()` turns the result of diff() into a more usable object. At this point, the expression returns `[Object: {"days": -<number-of-days>}]`. For example, on 9th March, `[Object: {"days": -291}]`.
- \* `days` uses JMESPath syntax to retrieve just the number of days from the object. For more information on using JMESPath with n8n, refer to our [JMESpath](#)(/code/cookbook/jmespath.md) documentation. This gives you the number of days to Christmas, as a negative number.
- \* `toString().substring(1)` turns the number into a string and removes the `-`.
- \* `+ " days to Christmas!"`: another string, with a `+` to join it to the previous string.

---

## Query JSON with JMESPath

JMESPath is a query language for JSON that you can use to extract and transform elements from a JSON document. For full details of how to use JMESPath, refer to the [JMESPath documentation](#).

### The `jmespath()` method

n8n provides a custom method, `jmespath()`. Use this method to perform a search on a JSON object using the JMESPath query language.

The basic syntax is:

```
== "JavaScript" js $jmespath(object, searchString) ==
"Python" python _jmespath(object, searchString)
```

To help understand what the method does, here is the equivalent longer JavaScript:

```
var jmespath = require('jmespath');
jmespath.search(object, searchString);
```

object is a JSON object, such as the output of a previous node. searchString is an expression written in the JMESPath query language. The [JMESPath Specification](#) provides a list of supported expressions, while their [Tutorial](#) and [Examples](#) provide interactive examples.

## Common tasks

This section provides examples for some common operations. More examples, and detailed guidance, are available in [JMESPath's own documentation](#).

When trying out these examples, you need to set the Code node **Mode** to **Run Once for Each Item**.

### Apply a JMESPath expression to a collection of elements with projections

From the [JMESPath projections documentation](#):

Projections are one of the key features of JMESPath. Use it to apply an expression to a collection of elements. JMESPath supports five kinds of projections:

- List Projections
- Slice Projections
- Object Projections
- Flatten Projections
- Filter Projections

The following example shows basic usage of list, slice, and object projections. Refer to the [JMESPath projections documentation](#) for detailed explanations of each projection type, and more examples.

Given this JSON from a webhook node:

```
[
 {
 "headers": {
 "host": "n8n.instance.address",
 ...
 },
 "params": {},
 "query": {},
 "body": {
 "people": [
 {
 "first": "James",
 "last": "Green"
 },
 {
 "first": "Jacob",
 "last": "Jones"
 },
 {
 "first": "Jayden",
 "last": "Smith"
 }
],
 "dogs": {
 "Fido": {
 "color": "brown",
 "age": 7
 },
 "Spot": {
 "color": "black and white",
 "age": 5
 }
 }
 }
 }]
```

Retrieve a [list](#) of all the people's first names:

==== “Expressions (JavaScript)”

```
```js  
{{$jmespath($json.body.people, "[*].first" )}}  
// Returns ["James", "Jacob", "Jayden"]
```

```

```
==== "Code node (JavaScript)"

```js
let firstNames = $jmespath($json.body.people, "[*].first" )
return {firstNames};
/* Returns:
[
  {
    "firstNames": [
      "James",
      "Jacob",
      "Jayden"
    ]
  }
]
*/
```

==== "Code node (Python)" python firstNames =
_jmespath(_json.body.people, "[*].first") return
{"firstNames":firstNames} """ Returns: [
"firstNames": [
 "James",
 "Jacob",
 "Jayden"
] }] """

```

Get a slice of the first names:

```
==== "Expressions (JavaScript)"
```

```

```js
{{$jmespath($json.body.people, "[::2].first")}}
// Returns ["James", "Jacob"]
```

==== "Code node (JavaScript)" js let firstTwoNames =
$jmespath($json.body.people, "[::2].first"); return
{firstTwoNames}; /* Returns: [
"firstNames": [
 "James",
 "Jacob",
 "Jayden"
] }] */ === "Code node
(Python)" python firstTwoNames = _jmespath(_json.body.people, "
[::2].first") return {"firstTwoNames":firstTwoNames} """
Returns: [
 {
 "firstTwoNames": [
 "James",
 "Jacob"
]
 }
] """

```

Get a list of the dogs' ages using object projections:

```
==== "Expressions (JavaScript)"
```

```

```js
{{$jmespath($json.body.dogs, "*.age")}}
// Returns [7,5]
```

==== "Code node (JavaScript)" js let dogsAges =
$jmespath($json.body.dogs, "*.age"); return {dogsAges}; /*
Returns: [
 {
 "dogsAges": [
 7,
 5
]
 }
] */ ===
"Code node (Python)" python dogsAges =
_jmespath(_json.body.dogs, "*.age") return {"dogsAges":
```

```
dogsAges} """ Returns: [{
"dogsAges": [7, 5]
}] """
```

## Select multiple elements and create a new list or object

Use [Multiselect](#) to select elements from a JSON object and combine them into a new list or object.

Given this JSON from a webhook node:

```
[
 {
 "headers": {
 "host": "n8n.instance.address",
 ...
 },
 "params": {},
 "query": {},
 "body": {
 "people": [
 {
 "first": "James",
 "last": "Green"
 },
 {
 "first": "Jacob",
 "last": "Jones"
 },
 {
 "first": "Jayden",
 "last": "Smith"
 }
],
 "dogs": {
 "Fido": {
 "color": "brown",
 "age": 7
 },
 "Spot": {
 "color": "black and white",
 "age": 5
 }
 }
 }
 }]
```

Use multiselect list to get the first and last names and create new lists containing both names:

```
==== "Expressions (JavaScript)"

```js  
{{$jmespath($json.body.people, "[].[first, last]")}}  
// Returns  
```  

==== "Code node (JavaScript)"
```

```

```js
let newList = $jmespath($json.body.people, "[].[first, last]");
return {newList};
/* Returns:
[
  {
    "newList": [
      [
        "James",
        "Green"
      ],
      [
        "Jacob",
        "Jones"
      ],
      [
        "Jayden",
        "Smith"
      ]
    ]
  }
]
*/
```
==== "Code node (Python)" python newList =
_jmespath(_json.body.people, "[].[first, last]") return
{"newList":newList} """ Returns: [
"newList": [["James",
"Green"], ["Jacob",
"Jones"],
["Jayden",
"Smith"]] """
]
```

```

An alternative to arrow functions in expressions

For example, generate some input data by returning the below code from the Code node:

```

return[
  {
    "json": {
      "num_categories": "0",
      "num_products": "45",
      "category_id": 5529735,
      "parent_id": 1407340,
      "pos_enabled": 1,
      "pos_favorite": 0,
      "name": "HP",
      "description": "",
      "image": ""
    }
  },
  {
    "json": {
      "num_categories": "0",
      "num_products": "86",
      "category_id": 5529740,
      "parent_id": 1407340,
      "pos_enabled": 1,
    }
  }
]
```

```

```
 "pos_favorite": 0,
 "name": "Lenovo",
 "description": "",
 "image": ""
 }
}
]
```

You could do a search like “find the item with the name Lenovo and tell me their category ID.”

```
{{ $jmespath($("Code").all(), "[? json.name=='Lenovo'].json.category_id") }}
```

---

## Examples using n8n's built-in methods and variables

n8n provides built-in methods and variables for working with data and accessing n8n data. This section provides usage examples.

### Related resources

- [Built-in methods and variables reference](#)
  - [Expressions](#)
  - [Code node](#)
- 

## execution

### execution.id

Contains the unique ID of the current workflow execution.

```
==== "JavaScript" js let executionId = $execution.id; ====
"Python" python executionId = _execution.id
```

### execution.resumeUrl

The webhook URL to call to resume a [waiting](#) workflow.

See the [Wait > On webhook call](#) documentation to learn more.

`execution.resumeUrl` is available in workflows containing a Wait node, along with a node that waits for a webhook response.

### execution.customData

This is only available in the Code node.

```

==== "JavaScript" ``js // Set a single piece of custom execution data
$execution.customData.set("key", "value");

// Set the custom execution data object
$execution.customData.setAll({"key1": "value1", "key2": "value2"})

// Access the current state of the object during the execution
var customData = $execution.customData.getAll()

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

==== "Python" ```python # Set a single piece of custom execution
data _execution.customData.set("key", "value");

# Set the custom execution data object
_execution.customData.setAll({"key1": "value1", "key2": "value2"})

# 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")
```

```

Refer to [Custom executions data](#) for more information.

---



---

## getWorkflowStaticData(type)

This gives access to the static workflow data.

You can save data directly in the workflow. This data should be small.

As an example: you can save a timestamp of the last item processed from an RSS feed or database. It will always return an object. Properties can then read, delete or set on that object. When the workflow execution succeeds, n8n checks automatically if the data has changed and saves it, if necessary.

There are two types of static data, global and node. Global static data is the same in the whole workflow. Every node in the workflow can access it. The node static data is unique to the node. Only the node that set it can retrieve it again.

Example with global data:

```

==== "JavaScript" ```javascript // Get the global workflow static data
const workflowStaticData = $getWorkflowStaticData('global');

// Access its data
const lastExecution = workflowStaticData.lastExecution;

// Update its data
workflowStaticData.lastExecution = new Date().getTime();

// Delete data
delete workflowStaticData.lastExecution;

```

```

```
==== "Python" ````python # Get the global workflow static data
workflowStaticData = _getWorkflowStaticData('global')

# Access its data
lastExecution = workflowStaticData.lastExecution

# Update its data
workflowStaticData.lastExecution = new Date().getTime()

# Delete data
delete workflowStaticData.lastExecution
```

```

Example with node data:

```

==== "JavaScript" ````js // Get the static data of the node const
nodeStaticData = $getWorkflowStaticData('node');

// Access its data
const lastExecution = nodeStaticData.lastExecution;

// Update its data
nodeStaticData.lastExecution = new Date().getTime();

// Delete data
delete nodeStaticData.lastExecution;
```

==== "Python" ````python # Get the static data of the node
nodeStaticData = _getWorkflowStaticData('node')

# Access its data
lastExecution = nodeStaticData.lastExecution

# Update its data
nodeStaticData.lastExecution = new Date().getTime()

# Delete data
delete nodeStaticData.lastExecution
```

```

## Templates and examples

---

## Retrieve linked items from earlier in the workflow

Every item in a node's input data links back to the items used in previous nodes to generate it. This is useful if you need to retrieve linked items from further back than the immediate previous node.

To access the linked items from earlier in the workflow, use ("<nodeName>").itemMatching(currentNodeInputIndex).

For example, consider a workflow that does the following:

1. The Customer Datastore node generates example data: json

```
{
 "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" }, ...
}
```
2. The Edit Fields node simplifies this data: json

```
[{
 "name": "Jay Gatsby" }, {
 "name": "José Arcadio Buendía" }, ...
}
```
3. The Code node restore the email address to the correct person:

```
json [{
 "name": "Jay Gatsby",
 "restoreEmail": "gatsby@west-egg.com" }, {
 "name": "José Arcadio Buendía", "restoreEmail":
 "jab@macondo.co" }, ...
]
```

The Code node does this using the following code:

```
==== "JavaScript" js for(let i=0; i<$input.all().length; i++) {
$input.all()[i].json.restoreEmail = $('Customer Datastore (n8n
training)').itemMatching(i).json.email; } return
$input.all(); ===="Python" ``python for i,item in
enumerate(_input.all()): input.all()[i].json.restoreEmail = ('Customer
Datastore (n8n training)').itemMatching(i).json.email

return _input.all();
````
```

You can view and download the example workflow from [n8n website | itemMatchin usage example](#).

("**<node-name>**").all(**branchIndex?: number, runIndex?: number**)

This gives access to all the items of the current or parent nodes. If you don't supply any parameters, it returns all the items of the current node.

Getting items

```
==== "JavaScript" ``js // Returns all the items of the given node and
current run let allItems = $("").all();

// Returns all items the node "IF" outputs (index: 0 which is Output
// "true" of its most recent run)
let allItems = $("IF").all();

// Returns all items the node "IF" outputs (index: 0 which is Output
// "true" of the same run as current node)
let allItems = $("IF").all(0, $runIndex);

// Returns all items the node "IF" outputs (index: 1 which is Output
// "false" of run 0 which is the first run)
let allItems = $("IF").all(1, 0);
```

```
````

==== "Python" ````python # Returns all the items of the given node
and current run allItems = _("").all();

Returns all items the node "IF" outputs (index: 0 which is Output
#"true" of its most recent run)
allItems =_("IF").all();

Returns all items the node "IF" outputs (index: 0 which is Output
#"true" of the same run as current node)
allItems =_("IF").all(0, _runIndex);

Returns all items the node "IF" outputs (index: 1 which is Output
#"false" of run 0 which is the first run)
allItems =_("IF").all(1, 0);
````
```

Accessing item data

Get all items output by a previous node, and log out the data they contain:

```
==== "JavaScript" typescript    previousNodeData = $("<node-
name>").all();      for(let i=0; i<previousNodeData.length; i++) {
  console.log(previousNodeData[i].json);      } ===="Python" python
previousNodeData = _("<node-name>").all();      for item in
previousNodeData:          # item is of type <class
'pyodide.ffi.JsProxy'>          # You need to convert it to a Dict
itemDict = item.json.to_py()          print(itemDict)
```

vars

vars contains all Variables for the active environment. It's read-only: you can access variables using vars, but must set them using the UI.

```
==== "JavaScript" js      // Access a variable    $vars.<variable-
name> ===="Python" python      # Access a variable    _vars.
<variable-name>
```

Expressions cookbook

This section contains examples and recipes for tasks you can do with expressions.

Related resources

- Built-in methods and variables reference
 - Expressions
-

Check incoming data

At times, you may want to check the incoming data. If the incoming data doesn't match a condition, you may want to return a different value. For example, you want to check if a variable from the previous node is empty and return a string if it's empty. Use the following code snippet to return `not found` if the variable is empty.

```
{{$json["variable_name"]? $json["variable_name"] :"not found"}}
```

The above expression uses the ternary operator. You can learn more about the ternary operator [here](#).

As an alternative, you can use the [nullish coalescing operator \(??\)](#) or the [logical or operator \(||\)](#):

```
{} $x ?? "default value" {}  
{} $x || "default value" {}
```

In either of the above two cases, the value of `$x` will be used if it's set to a non-null, non-false value. The string `default value` is the fallback value.

Expressions common issues

Here are some common errors and issues related to [expressions](#) and steps to resolve or troubleshoot them.

The 'JSON Output' in item 0 contains invalid JSON

This error occurs when you use JSON mode but don't provide a valid JSON object. Depending on the problem with the JSON object, the error sometimes display as The 'JSON Output' in item 0 does not contain a valid JSON object.

To resolve this, make sure that the code you provide is valid JSON:

- Check the JSON with a [JSON validator](#).
- Check that your JSON object doesn't reference undefined input data. This may occur if the incoming data doesn't always include the same fields.

Can't get data for expression

This error occurs when n8n can't retrieve the data referenced by an expression. Often, this happens when the preceding node hasn't been run yet.

Another variation of this may appear as `Referenced node is unexecuted`. In that case, the full text of this error will tell you the exact node that isn't executing in this format:

An expression references the node ‘<node-name>’, but it hasn’t been executed yet. Either change the expression, or re-wire your workflow to make sure that node executes first.

To begin troubleshooting, test the workflow up to the named node.

For nodes that use JavaScript or other custom code, you can check if a previous node has executed before trying to use its value by checking the following:

```
$(<node-name>).isExecuted
```

As an example, this JSON references the parameters of the input data. This error will display if you test this step without connecting it to another node:

```
{
  "my_field_1": {{ $input.params }}
}
```

Invalid syntax

This error occurs when you use an expression that has a syntax error.

For example, the expression in this JSON includes a trailing period, which results in an invalid syntax error:

```
{
  "my_field_1": "value",
  "my_field_2": {{ $('If').item.json. }}
```

To resolve this error, check your [expression syntax](#) to make sure they follow the expected format.

Code node cookbook

This section contains examples and recipes for tasks you can do with the Code node.

Related resources

- [Built-in methods and variables reference](#)
 - [Code node](#)
-

Get number of items returned by the previous node

To get the number of items returned by the previous node:

```
==== "JavaScript"
```

```
```js
```

```

if (Object.keys(items[0].json).length === 0) {
 return [
 {
 json: {
 results: 0,
 }
 }
]
}
return [
 {
 json: {
 results: items.length,
 }
 }
];
```

```

The output will be similar to the following.

```

```json
[
 {
 "results": 8
 }
]
```
==== "Python" python      if len(items[0].json) == 0:          return
[           {           "json": {           "results": 0,           }           }       ]   else:
[           {           "json": {           "results": items.length,           }           }       ]```
The output will be similar to the following.
```json
[
 {
 "results": 8
 }
]
```

```

Get the binary data buffer

The binary data buffer contains all the binary file data processed by a workflow. You need to access it if you want to perform operations on the binary data, such as:

- Manipulating the data: for example, adding column headers to a CSV file.
- Using the data in calculations: for example, calculating a hash value based on it.
- Complex HTTP requests: for example, combining file upload with sending other data formats.

You can access the buffer using n8n's `getBinaryDataBuffer()` function:

```
/*
 * itemIndex: number. The index of the item in the input data.
 * binaryPropertyName: string. The name of the binary property.
 * The default in the Read/Write File From Disk node is 'data'.
 */
let binaryDataBufferItem = await
this.helpers.getBinaryDataBuffer(itemIndex, binaryPropertyName);
```

For example:

```
let binaryDataBufferItem = await this.helpers.getBinaryDataBuffer(0,
'data');
// Returns the data in the binary buffer for the first input item
```

You should always use the `getBinaryDataBuffer()` function, and avoid using older methods of directly accessing the buffer, such as targeting it with expressions like `items[0].binary.data.data`.

Output to the browser console with `console.log()` or `print()` in the Code node

You can use `console.log()` or `print()` in the Code node to help when writing and debugging your code.

For help opening your browser console, refer to [this guide by Balsamiq](#).

`console.log (JavaScript)`

For technical information on `console.log()`, refer to the [MDN developer docs](#).

For example, copy the following code into a Code node, then open your console and run the node:

```
let a = "apple";
console.log(a);
```

`print (Python)`

For technical information on `print()`, refer to the [Real Python's guide](#).

For example, set your Code node **Language** to **Python**, copy the following code into the node, then open your console and run the node:

```
a = "apple"
print(a)
```

Handling an output of `[object Object]`

If the console displays `[object Object]` when you print, check the data type, then convert it as needed.

To check the data type:

```
print(type(myData))
```

JsProxy

If `type()` outputs `<class 'pyodide.ffi.JsProxy'>`, you need to convert the JsProxy to a native Python object using `to_py()`. This occurs when working with data in the n8n node data structure, such as node inputs and outputs. For example, if you want to print the data from a previous node in the workflow:

```
previousNodeData = _("<node-name>").all();
for item in previousNodeData:
    # item is of type <class 'pyodide.ffi.JsProxy'>
    # You need to convert it to a Dict
    itemDict = item.json.to_py()
    print(itemDict)
```

Refer to the Pyodide documentation on [JsProxy](#) for more information on this class.

Examples using n8n's HTTP Request node

The HTTP Request node is one of the most versatile nodes in n8n. Use this node to make HTTP requests to query data from any app or service with a REST API.

Refer to [HTTP Request](#) for information on node settings.

Related resources

- [HTTP Request](#)
 - [Built-in methods and variables reference](#)
 - [Expressions](#)
-

Pagination in the HTTP Request node

The HTTP Request node supports pagination. This page provides some example configurations, including using the [HTTP node variables](#).

Refer to [HTTP Request](#) for more information on the node.

-8<- “_snippets/integrations/builtin/core-nodes/http/pagination-api-differences.md”

Enable pagination

In the HTTP Request node, select **Add Option > Pagination**.

Use a URL from the response to get the next page using \$response

If the API returns the URL of the next page in its response:

1. Set **Pagination Mode** to **Response Contains Next URL**. n8n displays the parameters for this option.
2. In **Next URL**, use an expression to set the URL. The exact expression depends on the data returned by your API. For example, if the API includes a parameter called `next-page` in the response body: `javascript {{ $response.body["next-page"] }}`

Get the next page by number using \$pageCount

If the API you're using supports targeting a specific page by number:

1. Set **Pagination Mode** to **Update a Parameter in Each Request**.
2. Set **Type** to **Query**.
3. Enter the **Name** of the query parameter. This depends on your API and is usually described in its documentation. For example, some APIs use a query parameter named `page` to set the page. So **Name** would be `page`.
4. Hover over **Value** and toggle **Expression** on.
5. Enter `{{ $pageCount + 1 }}`

`$pageCount` is the number of pages the HTTP Request node has fetched. It starts at zero. Most API pagination counts from one (the first page is page one). This means that adding `+1` to `$pageCount` means the node fetches page one on its first loop, page two on its second, and so on.

Navigate pagination through body parameters

If the API you're using allows you to paginate through the body parameters:

1. Set the HTTP Request Method to **POST**
2. Set **Pagination Mode** to **Update a Parameter in Each Request**.
3. Select **Body** in the **Type** parameter.
4. Enter the **Name** of the body parameter. This depends on the API you're using. `page` is a common key name.
5. Hover over **Value** and toggle **Expression** on.
6. Enter `{{ $pageCount + 1 }}`

Set the page size in the query

If the API you're using supports choosing the page size in the query:

1. Select **Send Query Parameters** in main node parameters (this is

the parameters you see when you first open the node, not the settings within options).

2. Enter the **Name** of the query parameter. This depends on your API. For example, a lot of APIs use a query parameter named `limit` to set page size. So **Name** would be `limit`.
 3. In **Value**, enter your page size.
-

Advanced AI

Build AI functionality using n8n: from creating your own chat bot, to using AI to process documents and data from other sources.

- **Get started**

Work through the short tutorial to learn the basics of building AI workflows in n8n.

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

- **Use a Starter Kit**

Try n8n's Self-hosted AI Starter Kit to quickly start building AI workflows.

[:octicons-arrow-right-24: Self-hosted AI Starter Kit](#)

- **Explore examples and concepts**

Browse examples and workflow templates to help you build. Includes explanations of important AI concepts.

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

- **How n8n uses LangChain**

Learn more about how n8n builds on LangChain.

[:octicons-arrow-right-24: LangChain in n8n](#)

- **Browse AI templates**

Explore a wide range of AI workflow templates on the n8n website.

[:octicons-arrow-right-24: AI workflows on n8n.io](#)

Related resources

Related documentation and tools.

Node types

This feature uses Cluster nodes: groups of root and sub nodes that work together.

-8<- “_snippets/integrations/builtin/cluster-nodes/cluster-nodes-summary.md”

Workflow templates

You can browse [workflow templates](#) in-app or on the n8n website [Workflows page](#).

Refer to [Templates](#) for information on accessing templates in-app.

Chat trigger

Use the [n8n Chat Trigger](#) to trigger a workflow based on chat interactions.

Chatbot widget

n8n provides a chatbot widget that you can use as a frontend for AI-powered chat workflows. Refer to the [@n8n/chat npm page](#) for usage information.

AI Workflow Builder

AI Workflow Builder enables you to create, refine, and debug workflows using natural language descriptions of your goals.

It handles the entire workflow construction process, including node selection, placement, and configuration, thereby reducing the time required to build functional workflows.

For details of pricing and availability of AI Workflow Builder, see [n8n Plans and Pricing](#).

Working with the builder

1. **Describe your workflow:** Either select an example prompt or describe your requirements in natural language.
2. **Monitor the build:** The builder provides real-time feedback through several phases.
3. **Review and refine the generated workflow:** Review required credentials and other parameters. Refine the workflow using prompts.

[Image: ai-workflow-builder.png]

Commands you can run in the builder

- `/clear`: Clears the context for the LLM and lets you start from scratch

Understanding credits

How credits work

Each time you send a message to the builder asking it to create or modify a workflow, that counts as one **interaction**, which is worth one **credit**.

✓ Counts as an interaction

- Sending a message to create a new workflow
- Asking the builder to modify an existing workflow
- Clicking the **Execute and refine** button in the builder window after a workflow is built

✗ Does NOT count as an interaction

- Messages that fail or produce generation errors
- Requests you manually stop by clicking the stop button

Getting more credits

If you've used your monthly limit, you can upgrade to a higher plan.

For details on plans and pricing, see [n8n Plans and Pricing](#).

AI model and data handling

The following data are sent to the LLM:

- Text prompts that you provide to create, refine, or debug the workflow
- Node definitions, parameters, and connections and the current workflow definition.
- Any mock execution data that is loaded when using the builder

The following data are not sent:

- Details of any credentials you use
- Past executions of the workflow

Accessing n8n MCP server

Connect supported MCP clients to your n8n workflows through n8n's built-in MCP server.

The server allows clients such as Lovable or Claude Desktop to connect securely to an n8n instance. Once connected, these clients can:

- Search within workflows marked as available in MCP
- Retrieve metadata and trigger information for workflows
- Trigger and run exposed workflows

Difference between instance-level MCP access and MCP Server Trigger node

Instance-level MCP access lets you create one connection per n8n instance, use centralized authentication, and choose which workflows to enable for access. Enabled workflows are easy to find and run without extra setup for each workflow.

In comparison, you configure an MCP Server Trigger node inside a single workflow. This node exposes tools only from that workflow, a useful approach when you want to craft a specific MCP server behavior within one workflow.

Key considerations when using instance-level MCP access

- It isn't a way to build or edit workflows from an AI client; authoring remains in n8n.
- It doesn't provide blanket exposure to all workflows in your instance. You must enable MCP at the instance level and then enable each workflow individually.
- It's not scoped to each MCP client. Any connected client sees all workflows you've enabled for MCP access.

Enabling MCP access

For Cloud and self-hosted instances

1. Navigate to **Settings > MCP Access**
2. Toggle **Enable MCP access** (requires instance owner or admin permissions).

[Image: enable-mcp-access.png]

Once enabled, you'll see:

- Connection instructions
- List of workflows exposed to MCP clients

To disable: Toggle the switch off.

For self-hosted: Complete disablement

To remove the feature entirely, set the environment variable:

`N8N_DISABLED_MODULES=mcp`

This action removes MCP endpoints and hides all related UI elements.

Setting up MCP authentication

The **How to connect** section on the **MCP Access** page provides two authentication methods for MCP clients:

- **OAuth2**
- **Access Token**

Using OAuth2

Copy your instance server URL from the **OAuth** tab and use it to configure your MCP client. After connecting, the client will redirect you to n8n to authorize access.

Revoking client access

To revoke access for connected MCP clients:

1. Navigate to **Settings > MCP Access**.
2. Switch to the **OAuth** tab in the **How to connect** section. You should see a table of connected clients in the **Connected OAuth clients** section.
3. Use the action menu in each client's row to revoke access for specific clients.

Using Access Token

Use your instance server URL and your personal MCP Access Token from the **Access Token** tab on the settings page.

When you first visit the **MCP Access page**, n8n automatically generates a personal MCP Access Token tied to your user account.

Rotating your token

If you lose your token or need to rotate it:

1. Navigate to **Settings > MCP Access**.
2. Make sure you are on the **Access Token** tab in the **How to connect** section.
3. Generate a new token.
n8n revokes the previous token when you generate a new one.
4. Update all connected MCP clients with the new value.

Exposing workflows to MCP clients

Workflow eligibility

In order for a workflow to be available to MCP clients, it must meet the following criteria:

1. Be active
2. Contain one of the following trigger nodes:
 - o Webhook
 - o Schedule
 - o Chat
 - o Form

By default, no workflows are visible to MCP clients. You must explicitly enable access.

Enabling access

Option 1: From the workflow editor

1. Open the workflow.
2. Go to **Settings**.
3. Toggle **Available in MCP**.

Option 2: From the workflows list

1. Go to **Workflows**.
2. Open the menu on a workflow card.
3. Select **Enable MCP access**.

Managing access

The **MCP Access settings page** shows all workflows available to MCP clients. From this list you can:

- Open a workflow directly
- Revoke access using the action menu (or use **Disable MCP access** from the workflow card menu)

Workflow descriptions

To help MCP clients identify workflows, you can add free-text descriptions as follows:

1. Open the workflow.
2. Click the pencil icon next to the workflow name.
3. Enter your description in the **Description** field.

[Image: mcp-access-workflow-descriptions.png]

Executing workflows through MCP clients

MCP clients can execute eligible workflows on your request. When a client triggers a workflow, it runs as usual in n8n, and you can monitor its execution in the **Executions** list. Once the execution is complete, the MCP client will retrieve the results.

Providing input data

If a workflow requires input data, the MCP client must provide that data when triggering the workflow. The workflow's trigger node determines the schema of the input data:

1. **Webhook trigger**: The MCP client will look for hints in workflow contents and its description. It's up to the workflow author to provide enough information for the client to generate valid input data.
2. **Schedule trigger**: No input data is required.
3. **Chat trigger**: Chat input format is determined by the chat node configuration.
4. **Form trigger**: Form fields are determined by the form node configuration.

Workflow timeouts

n8n enforces a 5-minute timeout for workflow executions triggered by MCP clients. If a workflow doesn't finish in time, n8n stops the execution and sends an error to the MCP client, ignoring any timeout you set in the workflow settings for MCP-triggered executions.

Limitations

- If there are multiple supported triggers in a workflow, MCP clients may only be able to use one (first one) of them to trigger the workflow.
- Executing workflows with multi-step forms or any kind of human-in-the-loop interactions isn't supported.
- Binary input data isn't supported. MCP clients can only provide text-based inputs for your workflows.

Examples

Connecting Lovable to n8n MCP server

1. Configure MCP Server in Lovable (OAuth).
 - Navigate to your workspace **Settings > Integrations**.
 - In the **MCP Servers** section, find **n8n** and click **Connect**.
 - Enter your n8n server URL (shown on the **MCP Access** page).
 - Save the connection. If successful, n8n redirects you to authorize Lovable.
2. Verify connectivity.
 - Once connected, Lovable can query for workflows with MCP access enabled.
 - **Example:** Asking Lovable to build a workflow UI that lists users and allows deleting them.

Connecting Claude Desktop to n8n MCP server

Using OAuth2

1. Navigate to **Settings > Connectors** in Claude Desktop.
2. Click on **Add custom connector**.
3. Enter the following details:
 - **Name:** n8n MCP
 - **Remote MCP Server URL:** Your n8n base URL (shown on the **MCP Access** page)
4. Save the connector.
5. When prompted, authorize Claude Desktop to access your n8n instance.

Using Access Token

Add the following entry to your `claude_desktop_config.json` file:

```
{  
  "mcpServers": {  
    "n8n-mcp": {  
      "command": "npx",  
      "args": [
```

```

        "-y",
        "supergateway",
        "--streamableHttp",
        "https://<your-n8n-domain>/mcp-server/http",
        "--header",
        "authorization:Bearer <YOUR_N8N_MCP_TOKEN>"
    ]
}
}
}

```

Here, replace:

- <your-n8n-domain>: Your n8n base URL (shown on the **MCP Access page**)
- <YOUR_N8N_MCP_TOKEN>: Your generated token

Connecting Claude Code to n8n MCP server

Use the following CLI command:

```
claude mcp add --transport http n8n-mcp https://<your-n8n-
domain>/mcp-server/http \
--header "Authorization: Bearer <YOUR_N8N_MCP_TOKEN>"
```

Alternatively, add the following entry to your `claude.json` file:

```

{
  "mcpServers": {
    "n8n-local": {
      "type": "http",
      "url": "https://<your-n8n-domain>/mcp-server/http",
      "headers": {
        "Authorization": "Bearer <YOUR_N8N_MCP_TOKEN>"
      }
    }
  }
}
```

Here, replace:

- <your-n8n-domain>: Your n8n base URL (shown on the **MCP Access page**)
- <YOUR_N8N_MCP_TOKEN>: Your generated token

Connecting Codex CLI to n8n MCP server

Add the following entry to your `~/.codex/config.toml` file:

```
[mcp_servers.n8n_mcp]
command = "npx"
args = [
  "-y",
  "supergateway",
  "--streamableHttp",
  "https://<your-n8n-domain>/mcp-server/http",
  "--header",
  "authorization:Bearer <YOUR_N8N_MCP_TOKEN>"
]
```

Here, replace:

- <your-n8n-domain>: Your n8n base URL (shown on the **MCP Access** page)
- <YOUR_N8N_MCP_TOKEN>: Your generated token

Troubleshooting

If you encounter issues connecting MCP clients to your n8n instance, consider the following:

- Ensure that your n8n instance is publicly accessible if you are using cloud-based MCP clients.
 - Verify that the MCP access is enabled in n8n settings.
 - Check that the workflows you want to access are marked as available in MCP.
 - Confirm that the authentication method (OAuth2 or Access Token) is correctly configured in your MCP client.
 - Review n8n server logs for any error messages related to MCP connections.
 - If you are using desktop MCP clients, make sure you have the latest [Node.js](#) version installed.
-

Build an AI chat agent with n8n

Welcome to the introductory tutorial for building AI workflows with n8n. Whether you have used n8n before, or this is your first time, we will show you how the building blocks of AI workflows fit together and construct a working AI-powered chat agent which you can easily customize for your own purposes.

[Image: “Screenshot of the completed workflow”]

Many people find it easier to take in new information in video format. This tutorial is based on one of n8n’s popular videos, linked below. Watch the video or read the steps here, or both!

What you will need

- **n8n**: For this tutorial we recommend using the [n8n cloud](#) service - there is a free trial for new users! For a self hosted service, refer to the [installation pages](#).
- **Credentials for a chat model**: This tutorial uses OpenAI, but you can easily use DeepSeek, Google Gemini, Groq, Azure, and others (see the [sub-nodes documentation](#) for more).

What you will learn

- AI concepts in n8n
- How to use the AI Agent node
- Working with Chat input
- Connecting with AI models
- Customising input
- Observing the conversation
- Adding persistence

AI concepts in n8n

If you're already familiar with AI, feel free to skip this section. This is a basic introduction to AI concepts and how they can be used in n8n workflows.

An AI agent builds on Large Language Models (LLMs), which generate text based on input by predicting the next word. While LLMs only process input to produce output, AI agents add goal-oriented functionality. They can use tools, process their outputs, and make decisions to complete tasks and solve problems.

In n8n, the AI agent is represented as a node with some extra connections.

| Feature | LLM | AI Agent |
|---------------------|----------------------------|------------------------------------|
| Core Capability | Text generation | Goal-oriented task completion |
| Decision-Making | None | Yes |
| Uses Tools/APIs | No | Yes |
| Workflow Complexity | Single-step | Multi-step |
| Scope | Generates language | Performs complex, real-world tasks |
| Example | LLM generating a paragraph | An agent scheduling an appointment |

By incorporating the AI agent as a node, n8n can combine AI-driven steps with traditional programming for efficient, real-world workflows. For instance, simpler tasks, like validating an email address, do not require AI, whereas a complex tasks, like processing the *content* of an email or dealing with multimodal inputs (e.g., images, audio), are excellent uses of an AI agent.

1. Create a new workflow

-8<- “_snippets/try-it-out/new-workflow.md”

2. Add a trigger node

Every workflow needs somewhere to start. In n8n these are called 'trigger nodes'. For this workflow, we want to start with a chat node.

1. Select **Add first step** or press ++tab++ to open the node menu.
2. Search for **Chat Trigger**. n8n shows a list of nodes that match the search.
3. Select **Chat Trigger** to add the node to the canvas. n8n opens the node.
4. Close the node details view (Select **Back to canvas**) to return to the canvas.

??? explanation “More about the Chat Trigger node...” The trigger node generates output when there is an event causing it to trigger. In this case we want to be able to type in text to cause the workflow to run. In production, this trigger can be hooked up to a public chat interface as provided by n8n or embedded into another website. To start this simple workflow we will just use the built-in local chat interface to communicate, so no further setup is required.

3. Add an AI Agent Node

The AI Agent node is the core of adding AI to your workflows.

1. Select the **Add node** Image: Add node icon connector on the trigger node to bring up the node search.
2. Start typing “AI” and choose the **AI agent** node to add it.
3. The editing view of the **AI agent** will now be displayed.
4. There are some fields which can be changed. As we’re using the **Chat Trigger** node, the default setting for the source and specification of the prompt don’t need to be changed.

4. Configure the node

AI agents require a chat model to be attached to process the incoming prompts.

1. Add a chat model by clicking the plus Image: Add node icon button underneath the **Chat Model** connection on the **AI Agent** node (it’s the first connection along the bottom of the node).
2. The search dialog will appear, filtered on ‘Language Models’. These are the models with built-in support in n8n. For this tutorial we will use **OpenAI Chat Model**.
3. Selecting the **OpenAI Chat model** from the list will attach it to the **AI Agent** node and open the node editor. One of the parameters which can be changed is the ‘Model’. Note that for the basic OpenAI accounts, only the ‘gpt-4o-mini’ model is allowed.

??? explanation “Which chat model?” As mentioned earlier, the LLM is the component which generates the text according to a prompt it is given. LLMs have to be created and trained, usually an intensive process. Different LLMS may have different capabilities or specialties, depending on the data they were trained with.

5. Add credentials (if needed)

In order for n8n to communicate with the chat model, it will need some credentials (login data giving it access to an account on a different online service). If you already have credentials set up for OpenAI, these should appear by default in the credentials selector. Otherwise you can use the Credentials selector to help you add a new credential.

[Image: image showing the credentials dialog for OpenAI]

1. To add a new credential, click on the text which says ‘Select credential’. An option to add a new credential will appear [Image: Screenshot showing create a new credential button]
2. This credential just needs an API key. When adding credentials of any type, check the text to the right-hand side. In this case it has a handy link to take you straight to your OpenAI account to retrieve the API key.
3. The API key is just one long string. That’s all you need for this particular credential. Copy it from the OpenAI website and paste it into the **API key** section.

??? explanation “Keeping your credentials safe” 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.

6. Test the node

Now that the node is connected to the **Chat Trigger** and a chat model, we can test this part of the workflow.

1. Click on the ‘Chat’ button near the bottom of the canvas. This opens up a local chat window on the left and the AI agent logs on the right.
2. Type in a message and press ++enter++. You will now see the response from the chat model appear below your message.
3. The log window displays the inputs to and outputs from the AI Agent. [Image: image showing a chat session in progress]

??? explanation “Accessing the logs...” You can access the logs for the AI node even when you aren’t using the chat interface. Open up the **AI Agent** node and click on the **Logs** tab in the right hand panel. [Image: screenshot showing the Logs tab in the AI Agent]

7. Changing the prompt

The logs in the previous step reveal some extra data - the system prompt. This is the default message that the **AI Agent** primes the chat model with. From the log you can see this is set to “You are a helpful assistant”. We can however change this prompt to alter the behavior of the chat model.

1. Open the **AI Agent** node. In the bottom of the panel is a section labeled ‘Options’ and a selector labeled ‘Add Option’. Use this to select ‘System message’
2. The system message is now displayed. This is the same priming prompt we noticed before in the logs. Change the prompt to something else to prime the chat model in a different way. You could try something like “You are a brilliant poet who always replies in rhyming couplets” for example.

3. Close the node and return to the chat window. Repeat your message and notice how the output has changed. [Image: image showing changed text for chat, now it rhymes; if you can believe that]

8. Adding persistence

The chat model is now giving us useful output, but there is something wrong with it which will become apparent when you try to have a conversation.

1. Use the chat and tell the chat model your name, for example "Hi there, my name is Nick".
2. Wait for the response, then type the message "What's my name?". The AI will not be able to tell you, however apologetic it may seem. The reason for this is we are not saving the context. The AI Agent has no memory. [Image: image showing a conversation illustrating the above]
3. In order to remember what has happened in the conversation, the AI Agent needs to preserve context. We can do this by adding memory to the **AI Agent** node. On the canvas click on the Image: Add node icon on the bottom of the **AI Agent** node labeled "Memory".
4. From the panel which appears, select "Simple Memory". This will use the memory from the instance running n8n, and is usually sufficient for simple usage. The default value of 5 interactions should be sufficient here, but remember where this option is if you may want to change it later.
5. Repeat the exercise of having a conversation above, and see that the AI Agent now remembers your name.

9. Saving the workflow

Before we leave the workflow editor, remember to save the workflow or all your changes will be lost.

1. Click on the "Save" button in the top right of the editor window. Your workflow will now be saved and you can return to it later to chat again or add new features.

Congratulations!

You have taken your first steps in building useful and effective workflows with AI. In this tutorial we have investigated the basic building blocks of an AI workflow, added an **AI Agent** and a chat model, and adjusted the prompt to get the kind of output we wanted. We also added memory so the chat could retain context between messages.

Next steps

Now you have seen how to create a basic AI workflow, there are plenty of resources to build on that knowledge and plenty of examples to give you ideas of where to go next:

- Learn more about AI concepts and view examples in [Examples and concepts](#).
 - Browse AI [Workflow templates](#).
 - Find out how to [enhance the AI agent with tools](#).
-

RAG in n8n

What is RAG

Retrieval-Augmented Generation (RAG) is a technique that improves AI responses by combining language models with external data sources. Instead of relying solely on the model's internal training data, RAG systems retrieve relevant documents to ground responses in up-to-date, domain-specific, or proprietary knowledge. RAG workflows typically rely on vector stores to manage and search this external data efficiently.

What is a vector store?

A vector store is a special database designed to store and search high-dimensional vectors: numerical representations of text, images, or other data. When you upload a document, the vector store splits it into chunks and converts each chunk into a vector using an embedding model.

You can query these vectors using similarity searches, which construct results based on *semantic meaning*, rather than keyword matches. This makes vector stores a powerful foundation for RAG and other AI systems that need to retrieve and reason over large sets of knowledge.

How to use RAG in n8n

Inserting data into your vector store

Before your agent can access custom knowledge, you need to upload that data to a vector store:

1. Add the nodes needed to fetch your source data.
2. Insert a **Vector Store** node (e.g. the [Simple Vector Store](#)) and choose the **Insert Documents** operation.
3. Select an **embedding model**, which converts your text into vector embeddings. Consult the FAQ for more information on [choosing the right embedding model](#).
4. Add a Default Data Loader node, which splits your content into chunks. You can use the default settings or define your own chunking strategy:
 - **Character Text Splitter:** splits by character length.
 - **Recursive Character Text Splitter:** recursively splits by

Markdown, HTML, code blocks or simple characters
(recommended for most use cases).

- **Token Text Splitter:** splits by token count.
5. (Optional) Add **metadata** to each chunk to enrich the context and allow better filtering later.

Querying your data

You can query the data in two main ways: using an agent or directly through a node.

Using agents

1. Add an [agent](#) to your workflow.
2. Add the vector store as a **tool** and give it a **description** to help the agent understand when to use it:
 - Set the **limit** to define how many chunks to return.
 - Enable **Include Metadata** to provide extra context for each chunk.
3. Add the same **embedding model** you used when inserting the data.

Using the node directly

1. Add your vector store node to the canvas and choose the **Get Many** operation.
2. Enter a query or prompt:
 - Set a **limit** for how many chunks to return.
 - Enable **Include Metadata** if needed.

FAQs

How do I choose the right embedding model?

The right embedding model differs from case to case.

In general, smaller models (for example, `text-embedding-ada-002`) are faster and cheaper and thus ideal for short, general-purpose documents or lightweight RAG workflows. Larger models (for example, `text-embedding-3-large`) offer better semantic understanding. These are best for long documents, complex topics, or when accuracy is critical.

What is the best text splitting for my use case?

This again depends a lot on your data:

- Small chunks (for example, 200 to 500 tokens) are good for fine-grained retrieval.
- Large chunks may carry more context but can become diluted or noisy.

Using the right overlap size is important for the AI to understand the context of the chunk. That's also why using the Markdown or Code Block splitting can often help to make chunks better.

Another good approach is to add more context to it (for example, about the document where the chunk came from). If you want you can read more about this, you can check out [this great article from Anthropic](#).

LangChain in n8n

n8n provides a collection of nodes that implement LangChain's functionality. The LangChain nodes are configurable, meaning you can choose your preferred agent, LLM, memory, and so on. 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.

- [Learning resources](#): n8n's documentation for LangChain assumes you're familiar with AI and LangChain concepts. This page provides links to learning resources.
 - [LangChain concepts and features in n8n](#): how n8n represents LangChain concepts and features.
-

LangChain concepts in n8n

This page explains how LangChain concepts and features map to n8n nodes.

This page includes lists of the LangChain-focused nodes in n8n. You can use any n8n node in a workflow where you interact with LangChain, to link LangChain to other services. The LangChain features uses n8n's [Cluster nodes](#).

Trigger nodes

[Chat Trigger](#)

Cluster nodes

-8<- “_snippets/integrations/builtin/cluster-nodes/cluster-nodes-summary.md”

Root nodes

Each cluster starts with one [root node](#).

Chains

A [chain](#) is a series of LLMs, and related tools, linked together to support functionality that can't be provided by a single LLM alone.

Available nodes:

- [Basic LLM Chain](#)
- [Retrieval Q&A Chain](#)

- [Summarization Chain](#)
- [Sentiment Analysis](#)
- [Text Classifier](#)

Learn more about [chaining in LangChain](#).

Agents

An [agent](#){ data-preview} has access to a suite of tools, and determines which ones to use depending on the user input. Agents can use multiple tools, and use the output of one tool as the input to the next. [Source](#)

Available nodes:

- [Agent](#)

Learn more about [Agents in LangChain](#).

Vector stores

[Vector stores](#) store embedded data, and perform vector searches on it.

- [Simple Vector Store](#)
- [PGVector Vector Store](#)
- [Pinecone Vector Store](#)
- [Odrant Vector Store](#)
- [Supabase Vector Store](#)
- [Zep Vector Store](#)

Learn more about [Vector stores in LangChain](#).

Miscellaneous

Utility nodes.

[LangChain Code](#): import LangChain. This means if there is functionality you need that n8n hasn't created a node for, you can still use it.

Sub-nodes

Each root node can have one or more [sub-nodes](#) attached to it.

Document loaders

Document loaders add data to your chain as documents. The data source can be a file or web service.

Available nodes:

- [Default Document Loader](#)
- [GitHub Document Loader](#)

Learn more about [Document loaders in LangChain](#).

Language models

[LLMs \(large language models\)](#) are programs that analyze datasets. They're the key element of working with AI.

Available nodes:

- [Anthropic Chat Model](#)
- [AWS Bedrock Chat Model](#)
- [Cohere Model](#)
- [Hugging Face Inference Model](#)
- [Mistral Cloud Chat Model](#)
- [Ollama Chat Model](#)
- [Ollama Model](#)
- [OpenAI Chat Model](#)

Learn more about [Language models in LangChain](#).

Memory

[Memory](#) retains information about previous queries in a series of queries. For example, when a user interacts with a chat model, it's useful if your application can remember and call on the full conversation, not just the most recent query entered by the user.

Available nodes:

- [Motorhead](#)
- [Redis Chat Memory](#)
- [Postgres Chat Memory](#)
- [Simple Memory](#)
- [Xata](#)
- [Zep](#)

Learn more about [Memory in LangChain](#).

Output parsers

Output parsers take the text generated by an LLM and format it to match the structure you require.

Available nodes:

- [Auto-fixing Output Parser](#)
- [Item List Output Parser](#)
- [Structured Output Parser](#)

Learn more about [Output parsers in LangChain](#).

Retrievers

- [Contextual Compression Retriever](#)
- [MultiQuery Retriever](#)
- [Vector Store Retriever](#)
- [Workflow Retriever](#)

Text splitters

Text splitters break down data (documents), making it easier for the LLM to process the information and return accurate results.

Available nodes:

- [Character Text Splitter](#)
- [Recursive Character Text Splitter](#)
- [Token Splitter](#)

n8n's text splitter nodes implements parts of [LangChain's text splitter API](#).

Tools

Utility [tools](#).

- [Calculator](#)
- [Code Tool](#)
- [SerpAPI](#)
- [Think Tool](#)
- [Vector Store Tool](#)
- [Wikipedia](#)
- [Wolfram|Alpha](#)
- [Workflow Tool](#)

Embeddings

[Embeddings](#) capture the “relatedness” of text, images, video, or other types of information. ([source](#))

Available nodes:

- [Embeddings AWS Bedrock](#)
- [Embeddings Cohere](#)
- [Embeddings Google PaLM](#)
- [Embeddings Hugging Face Inference](#)
- [Embeddings Mistral Cloud](#)
- [Embeddings Ollama](#)
- [Embeddings OpenAI](#)

Learn more about [Text embeddings in LangChain](#).

Miscellaneous

- [Chat Memory Manager](#)
-

LangChain learning resources

You don't need to know details about LangChain to use n8n, but it can be helpful to learn a few concepts. This pages lists some learning resources that people at n8n have found helpful.

The [LangChain documentation](#) includes introductions to key concepts and possible use cases. Choose the [LangChain | Python](#) or [LangChain | JavaScript](#) documentation for quickstarts, code examples, and API documentation. LangChain also provide [code templates](#) (Python only), offering ideas for potential use cases and common patterns.

[What Product People Need To Know About LangChain](#) provides a list of terminology and concepts, explained with helpful metaphors. Aimed at a wide audience.

If you prefer video, this [YouTube series by Greg Kamradt](#) works through the LangChain documentation, providing code examples as it goes.

n8n offers space to discuss LangChain on the [Discord](#). Join to share your projects and discuss ideas with the community.

Use LangSmith with n8n

[LangSmith](#) is a developer platform created by the LangChain team. You can connect your n8n instance to LangSmith to record and monitor runs in n8n, just as you can in a LangChain application.

Connect your n8n instance to LangSmith

1. [Log in to LangSmith](#) and get your API key.
2. Set the LangSmith environment variables:

| Variable | Value |
|--------------------------------|---|
| LANGCHAIN_ENDPOINT | " https://api.smith.langchain.com " |
| LANGCHAIN_TRACING_V2 | true |
| LANGCHAIN_API_KEY | Set this to your API key |
| LANGCHAIN_PROJECT | Optional project name (defaults to "default") |
| LANGCHAIN_CALLBACKS_BACKGROUND | true (asynchronous trace upload) |

Set the variables so that they're available globally in the environment where you host your n8n instance. You can do this in the same way as the rest of your general configuration.

3. Restart n8n.

For information on using LangSmith, refer to [LangSmith's documentation](#).

Overview

What are evaluations?

Evaluation is a crucial technique for checking that your AI workflow is reliable. It can be the difference between a flaky proof of concept and a solid production workflow. It's important both in the building phase and after deploying to production.

The foundation of evaluation is running a test dataset through your workflow. This dataset contains multiple test cases. Each test case contains a sample input for your workflow, and often includes the expected output(s) too.

Evaluation allows you to:

- **Test your workflow over a range of inputs** so you know how it performs on edge cases

- **Make changes with confidence** without inadvertently making things worse elsewhere
- **Compare performance** across different models or prompts

The following video explains what evaluations are, why they're useful, and how they work:

Why is evaluation needed?

AI models are fundamentally different than code. Code is deterministic and you can reason about it. This is difficult to do with LLMs, since they're black boxes. Instead, you must *measure* LLM output by running data through them and observing the output.

You can only build confidence that your model performs reliably after you have run it over multiple inputs that accurately reflect all the edge cases that it will have to deal with in production.

Two types of evaluation

Light evaluation (pre-deployment)

Building a clean, comprehensive dataset is hard. In the initial building phase, it often makes sense to generate just a handful of examples. These can be enough to iterate the workflow to a releasable state (or a proof of concept). You can visually compare the results to get a sense of the workflow's quality, without setting up formal metrics.

Metric-based evaluation (post-deployment)

Once you deploy your workflow, it's easier to build a bigger, more representative dataset from production executions. When you discover a bug, you can add the input that caused it to the dataset. When fixing the bug, it's important to run the whole dataset over the workflow again as a regression test to check that the fix hasn't inadvertently made something else worse.

Since there are too many test cases to check individually, evaluations measure the quality of the outputs using a metric, a numeric value representing a particular characteristic. This also allows you to track quality changes between runs.

Comparison of evaluation types

| | Light evaluation
(pre-deployment) | Metric-based
evaluation (post-deployment) |
|---|---|--|
| Performance improvements with each iteration | Large | Small |
| Dataset size | Small | Large |
| Dataset sources | Hand-generated
AI-generated
Other | Production executions
AI-generated
Other |

| | | |
|--------------------------|----------|--------------------|
| Actual outputs | Required | Required |
| Expected outputs | Optional | Required (usually) |
| Evaluation metric | Optional | Required |

Learn more

- Light evaluations: Perfect for evaluating your AI workflows against hand-selected test cases during development.
- Metric-based evaluations: Advanced evaluations to maintain performance and correctness in production by using scoring and metrics with large datasets.
- Tips and common issues: Learn how to set up specific evaluation use cases and work around common issues.

Light evaluations

What are light evaluations?

When building your workflow, you often want to test it with a handful of examples to get a sense of how it performs and make improvements. At this stage of workflow development, looking over workflow outputs for each example is often enough. The benefits of setting up more formal scoring or metrics don't yet justify the effort.

Light evaluation allows you to run the examples in a test dataset through your workflow one-by-one, writing the outputs back to your dataset. You can then examine those outputs next to each other, and visually compare them to the expected outputs (if you have them).

How it works

Light evaluations take place in the 'Editor' tab of your workflow, although you'll find instructions on how to set it up in the 'Evaluations' tab.

Steps:

1. Create a dataset
2. Wire the dataset up to the workflow
3. Write workflow outputs back to dataset
4. Run evaluation

The following explanation will use a sample workflow that assigns a category and priority to incoming support tickets.

[Image: Example AI workflow]

1. Create a dataset

Create a data table or Google Sheet with a handful of examples for your workflow. Your dataset should contain columns for:

- The workflow input

- (Optional) The expected or correct workflow output
- The actual output

Leave the actual output column or columns blank, since you'll be filling them during the evaluation.

[Image: Sample dataset for a support ticket classification workflow]

A [sample dataset](#) for the support ticket classification workflow.

2. Wire the dataset up to your workflow

Insert an evaluation trigger to pull in your dataset

Each time the [evaluation trigger](#) runs, it will output a single item representing one row of your dataset.

Clicking the 'Evaluate all' button to the left of the evaluation trigger will run your workflow multiple times in sequence, once for each row in your dataset. This is a special behavior of the evaluation trigger.

While wiring the trigger up, you often only want to run it once. You can do this by either:

- Setting the trigger's 'Max rows to process' to 1
- Clicking on the 'Execute node' button on the trigger (rather than the 'Evaluate all' button)

Wire the trigger up to your workflow

You can now connect the evaluation trigger to the rest of your workflow and reference the data that it outputs. At a minimum, you need to use the dataset's input column(s) later in the workflow.

If you have multiple triggers in your workflow you will need to [merge their branches together](#).

[Image: Connecting the evaluation trigger]

The support ticket classification workflow with the evaluation trigger added in and wired up.

3. Write workflow outputs back to dataset

To populate the output column(s) of your dataset when the evaluation runs:

- Insert the 'Set outputs' action of the [evaluation node](#)
- Wire it up to your workflow at a point after it has produced the outputs you're evaluating
- In the node's parameters, map the workflow outputs into the correct dataset column

[Image: Connecting the set outputs node]

The support ticket classification workflow with the 'set outputs' node added in and wired up.

4. Run evaluation

Click on the **Execute workflow** button to the left of the evaluation trigger. The workflow will execute multiple times, once for each row of the dataset:

[Image: Execute workflow button]

Review the outputs of each execution in the data table or Google Sheet, and examine the execution details using the workflow's 'executions' tab if you need to.

Once your dataset grows past a handful of examples, consider [metric-based evaluation](#) to get a numerical view of performance. See also [tips and common issues](#).

Metric-based evaluations

What are metric-based evaluations?

Once your workflow is ready for deployment, you often want to test it on more examples than [when you were building it](#).

For example, when production executions start to turn up edge cases, you want to add them to your test dataset so that you can make sure they're covered.

For large datasets like the ones built from production data, it can be hard to get a sense of performance just by eyeballing the results. Instead, you must measure performance. Metric-based evaluations can assign one or more scores to each test run, which you can compare to previous runs. Individual scores get rolled up to measure performance on the whole dataset.

This feature allows you to run evaluations that calculate metrics, track how those metrics change between runs and drill down into the reasons for those changes.

Metrics can be deterministic functions (such as the distance between two strings) or you can calculate them using AI. Metrics often involve checking how far away the output is from a *reference output* (also called ground truth). To do so, the dataset must contain that reference output. Some evaluations don't need this reference output though (for example, checking text for sentiment or toxicity).

How it works

1. Set up [light evaluation](#)
2. Add metrics to workflow
3. Run evaluation and view results

1. Set up light evaluation

Follow the [setup instructions](#) to create a dataset and wire it up to your workflow, writing outputs back to the dataset.

The following steps use the same support ticket classification workflow from the light evaluation docs:

[Image: Light evaluation workflow]

2. Add metrics to workflow

Metrics are dimensions used to score the output of your workflow. They often compare the actual workflow output with a reference output. It's common to use AI to calculate metrics, although it's sometimes possible to just use code. In n8n, metrics are always numbers.

You need to add the logic to calculate the metrics for your workflow, at a point after it has produced the outputs. You can add any reference outputs your metric uses as a column in your dataset. This makes sure they will be available in the workflow, since they will be output by the evaluation trigger.

Use the **Set Metrics** operation to calculate:

- **Correctness (AI-based)**: Whether the answer's meaning is consistent with a supplied reference answer. Uses a scale of 1 to 5, with 5 being the best.
- **Helpfulness (AI-based)**: Whether the response answers the given query. Uses a scale of 1 to 5, with 5 being the best.
- **String Similarity**: How close the answer is to the reference answer, measured character-by-character (edit distance). Returns a score between 0 and 1.
- **Categorization**: Whether the answer is an exact match with the reference answer. Returns 1 when matching and 0 otherwise.
- **Tools Used**: Whether the execution used tools or not. Returns a score between 0 and 1.

You can also add custom metrics. Just calculate the metrics within the workflow and then map them into an Evaluation node. Use the **Set Metrics** operation and choose **Custom Metrics** as the Metric. You can then set the names and values for the metrics you want to return.

For example:

- RAG document relevance: when working with a vector database, whether the documents retrieved are relevant to the question.

Calculating metrics can add latency and cost, so you may only want to do it when running an evaluation and avoid it when making a production execution. You can do this by putting the metric logic after a 'check if evaluating' operation.

[Image: Check if evaluating node]

3. Run evaluation and view results

Switch to the **Evaluations** tab on your workflow and click the **Run evaluation** button. An evaluation will start. Once the evaluation has finished, it will display a summary score for each metric.

You can see the results for each test case by clicking on the test run row. Clicking on an individual test case will open the execution that produced it (in a new tab).

Tips and common issues

Combining multiple triggers

If you have another trigger in the workflow already, you have two potential starting points: that trigger and the evaluation trigger. To make sure your workflow works as expected no matter which trigger executes, you will need to merge these branches together.

[Image: Merging trigger branches]

Logic to merge two trigger branches together so that they have the same data format and can be referenced from a single node.

To do so:

1. **Get the data format of the other trigger:**
 - Execute the other trigger.
 - Open it and navigate to the JSON view of its output pane.
 - Click the **copy** button on the right.
2. **Re-shape the evaluation trigger data to match:**
 - Insert an Edit Fields (Set) node after the evaluation trigger and connect them together.
 - Change its mode to **JSON**.
 - Paste your data into the 'JSON' field, removing the [and] on the first and last lines.
 - Switch the field type to **Expression**.
 - Map in the data from the trigger by dragging it from the input pane.
 - For strings, make sure to replace the entire value (including the quotes) and add .`toString()` to the end of the expression.
3. **Merge the branches using a 'No-op' node:** Insert a No-op node and wire both the other trigger and the Set node up to it. The 'No-op' node just outputs whatever input it receives.
4. **Reference the 'No-op' node outputs in the rest of the workflow:** Since both paths will flow through this node with the same format, you can be sure that your input data will always be there.

Avoiding evaluation breaking the chat

n8n's internal chat reads the output data of the last executed node in the workflow. After adding an evaluation node with the 'set outputs' operation, this data may not be in the expected format, or even contain the chat response.

[Image: Add second output branch]

The solution is to add an extra branch coming out of your agent. Lower branches execute later in n8n, which means any node you attach to this branch will execute last. You can use a no-op node here since it only needs to pass the agent output through.

Accessing tool data when calculating metrics

Sometimes you need to know what happened in executed sub-nodes of an agent, for example to check whether it executed a tool. You can't reference these nodes directly with expressions, but you can enable the **Return intermediate steps** option in the agent. This will add an extra output field called `intermediateSteps` which you can use in later nodes:

[Image: Enable return intermediate steps]

Multiple evaluations in the same workflow

You can only have one evaluation set up per workflow. In other words, you can only have one evaluation trigger per workflow.

Even so, you can still test different parts of your workflow with different evaluations by putting those parts in sub-workflows and evaluating each sub-workflow.

Dealing with inconsistent results

Metrics can often have noise: they may be different across evaluation runs of the exact same workflow. This is because the workflow itself may return different results, or any LLM-based metrics might have natural variation in them.

You can compensate for this by duplicating the rows of your dataset, so that each row appears more than once in the dataset. Since this means that each input will effectively be running multiple times, it will smooth out any variations.

Advanced AI examples and concepts

This section provides explanations of important AI concepts, and workflow templates that highlight those concepts, with explanations and configuration guides. The examples cover common use cases and highlight different features of advanced AI in n8n.

- **Agents and chains**

Learn about agents and chains in AI, including exploring key differences using the example workflow.

[:octicons-arrow-right-24: What's a chain in AI?](#)

[:octicons-arrow-right-24: What's an agent in AI?](#)

[:octicons-arrow-right-24: Demonstration of key differences between agents and chains](#)

- **Call n8n Workflow Tool**

Learn about tools in AI, then explore examples that use n8n workflows as custom tools to give your AI workflow access to more data.

[:octicons-arrow-right-24: What's a tool in AI?](#)
[:octicons-arrow-right-24: Chat with Google Sheets](#)
[:octicons-arrow-right-24: Call an API to fetch data](#)
[:octicons-arrow-right-24: Set up a human fallback](#)
[:octicons-arrow-right-24: Let AI specify tool parameters with \\$fromAI\(\)](#)

- **Vector databases**

Learn about [vector databases](#) in AI, along with related concepts including [embeddings](#) and [retrievers](#).

[:octicons-arrow-right-24: What's a vector database?](#)
[:octicons-arrow-right-24: Populate a Pinecone vector database from a website](#)

- **Memory**

Learn about [memory](#) in AI.

[:octicons-arrow-right-24: What's memory in AI?](#)

- **AI workflow templates**

You can browse AI templates, included community contributions, on the n8n website.

[:octicons-arrow-right-24: Browse all AI templates](#)

What's a chain in AI?

[Chains](#) bring together different components of AI to create a cohesive system. They set up a sequence of calls between the components. These components can include models and [memory](#) (though note that in n8n chains can't use memory).

Chains in n8n

n8n provides three chain nodes:

- [Basic LLM Chain](#): use to interact with an LLM, without any additional components.
- [Question and Answer Chain](#): can connect to a [vector store](#) using a retriever, or to an n8n workflow using the Workflow Retriever node. Use this if you want to create a workflow that supports asking questions about specific documents.
- [Summarization Chain](#): takes an input and returns a summary.

There's an important difference between chains in n8n and in other tools such as LangChain: none of the chain nodes support memory. This means they can't remember previous user queries. If you use LangChain to code an AI application, you can give your application memory. In n8n, if you need your workflow to support memory, use an agent. This is essential if you want users to be able to have a natural ongoing conversation with your app.

What's an agent in AI?

One way to think of an [agent](#) is as a [chain](#) that knows how to make decisions. Where a chain follows a predetermined sequence of calls to different AI components, an agent uses a language model to determine which actions to take.

Agents are the part of AI that act as decision-makers. They can interact with other agents and [tools](#). When you send a query to an agent, it tries to choose the best tools to use to answer. Agents adapt to your specific queries, as well as the prompts that configure their behavior.

Agents in n8n

n8n provides one Agent node, which can act as different types of agent depending on the settings you choose. Refer to the [Agent node documentation](#) for details on the available agent types.

When you execute a workflow containing an agent, the agent runs multiple times. For example, it may do an initial setup, followed by a run to call a tool, then another run to evaluate the tool response and respond to the user.

Demonstration of key differences between agents and chains

In this workflow you can choose whether your chat query goes to an [agent](#) or [chain](#). It shows some of the ways that agents are more powerful than chains.

Key features

This workflow uses:

- [Chat Trigger](#): start your workflow and respond to user chat interactions. The node provides a customizable chat interface.
- [Switch node](#): directs your query to either the agent or chain, depending on which you specify in your query. If you say “agent” it sends it to the agent. If you say “chain” it sends it to the chain.
- [Agent](#): the Agent node interacts with other components of the workflow and makes decisions about what [tools](#) to use.
- [Basic LLM Chain](#): the Basic LLM Chain node supports chatting with a connected LLM, but doesn’t support [memory](#) or tools.

Using the example

-8<- “_snippets/examples-color-key.md”

What's memory in AI?

Memory is a key part of AI chat services. The [memory](#) keeps a history of previous messages, allowing for an ongoing conversation with the AI, rather than every interaction starting fresh.

AI memory in n8n

To add memory to your AI workflow you can use either:

- [Simple Memory](#): stores a customizable length of chat history for the current session. This is the easiest to get started with.
- One of the memory services that n8n provides nodes for. These include:
 - [Motorhead](#)
 - [Redis Chat Memory](#)
 - [Postgres Chat Memory](#)
 - [Xata](#)
 - [Zep](#)

If you need to do advanced AI memory management in your workflows, use the [Chat Memory Manager](#) node.

-8<- “_snippets/integrations/builtin/cluster-nodes/langchain-sub-nodes/chat-memory-manager-purpose.md”

What's a tool in AI?

In AI, ‘tools’ has a specific meaning. Tools act like addons that your AI can use to access extra context or resources.

Here are a couple of other ways of expressing it:

Tools are interfaces that an agent can use to interact with the world ([source](#))

We can think of these tools as being almost like functions that your AI model can call ([source](#))

AI tools in n8n

n8n provides tool [sub-nodes](#) that you can connect to your [AI agent](#). As well as providing some popular tools, such as [Wikipedia](#) and [SerpAPI](#), n8n provides three especially powerful tools:

- [Call n8n Workflow Tool](#): use this to load any n8n workflow as a tool.
- [Custom Code Tool](#): write code that your agent can run.
- [HTTP Request Tool](#): make calls to fetch a website or data from an API.

The next three examples highlight the Call n8n Workflow Tool:

- [Chat with Google Sheets](#)
- [Call an API to fetch data](#)

- [Set up a human fallback](#)

You can also learn how to [let AI dynamically specify parameters for tools with the \\$fromAI\(\) function](#).

Chat with a Google Sheet using AI

Use n8n to bring your own data to AI. This workflow uses the [Chat Trigger](#) to provide the chat interface, and the [Call n8n Workflow Tool](#) to call a second workflow that queries Google Sheets.

Key features

This workflow uses:

- [Chat Trigger](#): start your workflow and respond to user chat interactions. The node provides a customizable chat interface.
- [Agent](#): the key piece of the AI workflow. The Agent interacts with other components of the workflow and makes decisions about what tools to use.
- [Call n8n Workflow Tool](#): plug in n8n workflows as custom tools. In AI, a tool is an interface the AI can use to interact with the world (in this case, the data provided by your workflow). The AI model uses the tool to access information beyond its built-in dataset.

Using the example

-8<- “_snippets/examples-color-key.md”

Call an API to fetch data

Use n8n to bring data from any [API](#) to your AI. This workflow uses the [Chat Trigger](#) to provide the chat interface, and the [Call n8n Workflow Tool](#) to call a second workflow that calls the API. The second workflow uses AI functionality to refine the API request based on the user's query.

Key features

This workflow uses:

- [Chat Trigger](#): start your workflow and respond to user chat interactions. The node provides a customizable chat interface.
- [Agent](#): the key piece of the AI workflow. The Agent interacts with other components of the workflow and makes decisions about what tools to use.
- [Call n8n Workflow Tool](#): plug in n8n workflows as custom tools. In AI, a tool is an interface the AI can use to interact with the world (in this case, the data provided by your workflow). The AI model uses the tool to access information beyond its built-in dataset.
- A [Basic LLM Chain](#) with an [Auto-fixing Output Parser](#) and

[Structured Output Parser](#) to read the user's query and set parameters for the API call based on the user input.

Using the example

-8<- “_snippets/examples-color-key.md”

Have a human fallback for AI workflows

This is a workflow that tries to answer user queries using the standard GPT-4 model. If it can't answer, it sends a message to Slack to ask for human help. It prompts the user to supply an email address.

This workflow uses the [Chat Trigger](#) to provide the chat interface, and the [Call n8n Workflow Tool](#) to call a second workflow that handles checking for email addresses and sending the Slack message.

Key features

This workflow uses:

- [Chat Trigger](#): start your workflow and respond to user chat interactions. The node provides a customizable chat interface.
- [Agent](#): the key piece of the AI workflow. The Agent interacts with other components of the workflow and makes decisions about what tools to use.
- [Call n8n Workflow Tool](#): plug in n8n workflows as custom tools. In AI, a tool is an interface the AI can use to interact with the world (in this case, the data provided by your workflow). It allows the AI model to access information beyond its built-in dataset.

Using the example

-8<- “_snippets/examples-color-key.md”

Let AI specify the tool parameters

When configuring [tools](#) connected to the Tools Agent, many parameters can be filled in by the AI model itself. The AI model will use the context from the task and information from other connected tools to fill in the appropriate details.

There are two ways to do this, and you can switch between them.

Let the model fill in the parameter

Each appropriate parameter field in the tool's editing dialog has an extra button at the end:

[Image: image showing stars icon to the right of parameter field]

On activating this button, the [AI Agent](#) will fill in the expression for you, with no need for any further user input. The field itself is filled in with a message indicating that the parameter has been defined automatically by the model.

If you want to define the parameter yourself, click on the 'X' in this box to revert to user-defined values. Note that the 'expression' field will now contain the expression generated by this feature, though you can now edit it further to add extra details as described in the following section.

Use the `$fromAI()` function

The `$fromAI()` function uses AI to dynamically fill in parameters for tools connected to the [Tools AI agent](#).

To use the `$fromAI()` function, call it with the required key parameter:

```
{} $fromAI('email') {}
```

The key parameter and other arguments to the `$fromAI()` function aren't references to existing values. Instead, think of these arguments as hints that the AI model will use to populate the right data.

For instance, if you choose a key called `email`, the AI Model will look for an email address in its context, other tools, and input data. In chat workflows, it may ask the user for an email address if it can't find one elsewhere. You can optionally pass other parameters like `description` to give extra context to the AI model.

Parameters

The `$fromAI()` function accepts the following parameters:

| Parameter | Type | Required? | Description |
|-------------|--------|--------------------|---|
| key | string | :white_check_mark: | A string representing the key or name of the argument. This must be between 1 and 64 characters in length and can only contain lowercase letters, uppercase letters, numbers, underscores, and hyphens. |
| description | string | :x: | A string describing the argument. |
| type | string | :x: | A string specifying the data type. Can be string, number, boolean, or json (defaults to string). The default value to |

| | | | |
|--------------|-----|-----|-----------------------|
| defaultValue | any | :x: | use for the argument. |
|--------------|-----|-----|-----------------------|

Examples

As an example, you could use the following `$fromAI()` expression to dynamically populate a field with a name:

```
$fromAI("name", "The commenter's name", "string", "Jane Doe")
```

If you don't need the optional parameters, you could simplify this as:

```
$fromAI("name")
```

To dynamically populate the number of items you have in stock, you could use a `$fromAI()` expression like this:

```
$fromAI("numItemsInStock", "Number of items in stock", "number", 5)
```

If you only want to fill in parts of a field with a dynamic value from the model, you can use it in a normal expression as well. For example, if you want the model to fill out the subject parameter for an e-mail, but always pre-fix the generated value with the string 'Generated by AI:', you could use the following expression:

```
Generated by AI: {{ $fromAI("subject") }}
```

Templates

You can see the `$fromAI()` function in action in the following [templates](#):

- [Angie, Personal AI Assistant with Telegram Voice and Text](#)
 - [Automate Customer Support Issue Resolution using AI Text Classifier](#)
 - [Scale Deal Flow with a Pitch Deck AI Vision, Chatbot and QDrant Vector Store](#)
-

What are vector databases?

Vector databases store information as numbers:

A vector database is a type of database that stores data as high-dimensional vectors, which are mathematical representations of features or attributes. ([source](#))

This enables fast and accurate similarity searches. With a vector database, instead of using conventional database queries, you can search for relevant data based on semantic and contextual meaning.

A simplified example

A vector database could store the sentence "n8n is a source-available automation tool that you can self-host", but instead of storing it as text, the vector database stores an array of dimensions (numbers

between 0 and 1) that represent its features. This doesn't mean turning each letter in the sentence into a number. Instead, the vectors in the vector database describe the sentence.

Suppose that in a vector store 0.1 represents automation tool, 0.2 represents source available, and 0.3 represents can be self-hosted. You could end up with the following vectors:

| Sentence | Vector (array of dimensions) |
|--|------------------------------|
| n8n is a source-available automation tool that you can self-host | [0.1, 0.2, 0.3] |
| Zapier is an automation tool | [0.1] |
| Make is an automation tool | [0.1] |
| Confluence is a wiki tool that you can self-host | [0.3] |

Demonstrating the power of similarity search

Qdrant provides [vector search demos](#) to help users understand the power of vector databases. The [food discovery demo](#) shows how a vector store can help match pictures based on visual similarities.

This demo uses data from Delivery Service. Users may like or dislike the photo of a dish, and the app will recommend more similar meals based on how they look. It's also possible to choose to view results from the restaurants within the delivery radius. ([source](#))

For full technical details, refer to the [Qdrant demo-food-discovery GitHub repository](#).

Embeddings, retrievers, text splitters, and document loaders

Vector databases require other tools to function:

- Document loaders and text splitters: document loaders pull in documents and data, and prepare them for [embedding](#). Document loaders can use text splitters to break documents into chunks.
- Embeddings: these are the tools that turn the data (text, images, and so on) into vectors, and back into raw data. Note that n8n only supports text embeddings.
- Retrievers: retrievers fetch documents from vector databases. You need to pair them with an embedding to translate the vectors back into data.

Populate a Pinecone vector database from a website

Use n8n to scrape a website, load the data into Pinecone, then query it using a chat workflow. This workflow uses the [HTTP node](#) to get website data, extracts the relevant content using the [HTML node](#), then uses the [Pinecone Vector Store node](#) to send it to Pinecone.

Key features

This workflow uses:

- [HTTP node](#): fetches website data.
- [HTML node](#): simplifies the data by extracting the main content from the page.
- [Pinecone Vector Store node](#) and [Embeddings OpenAI](#): transform the data into vectors and store it in Pinecone.
- [Chat Trigger](#) and [Question and Answer Chain](#) to query the vector database.

Using the example

-8<- “_snippets/examples-color-key.md”

n8n public REST API

Using n8n’s public [API](#), you can programmatically perform many of the same tasks as you can in the GUI. This section introduces n8n’s REST API, including:

- How to [authenticate](#)
- [Paginating results](#)
- Using the [built-in API playground](#) (self-hosted n8n only)
- [Endpoint reference](#)

n8n provides an [n8n API node](#) to access the API in your workflows.

Learn about REST APIs

The API documentation assumes you are familiar with REST APIs. If you’re not, these resources may be helpful:

- [KnowledgeOwl’s guide to working with APIs](#): a basic introduction, including examples of how to call REST APIs.
 - [IBM Cloud Learn Hub - What is an Application Programming Interface \(API\)](#): this gives a general, but technical, introduction to APIs.
 - [IBM Cloud Learn Hub - What is a REST API?](#): more detailed information about REST APIs.
 - [MDN web docs - An overview of HTTP](#): REST APIs work over HTTP and use HTTP verbs, or methods, to specify the action to perform.
-

API authentication

n8n uses API keys to authenticate API calls.

API Scopes

Users of [enterprise instances](#) can limit which resources and actions a key can access with scopes. API key scopes allow you specify the exact level of access a key needs for its intended purpose.

Non-enterprise API keys have full access to all the account's resources and capabilities.

Create an API key

1. Log in to n8n.
2. Go to **Settings > n8n API**.
3. Select **Create an API key**.
4. Choose a **Label** and set an **Expiration** time for the key.
5. If on an enterprise plan, choose the **Scopes** to give the key.
6. Copy **My API Key** and use this key to authenticate your calls.

Call the API using your key

Send the API key in your API call as a header named `X-N8N-API-KEY`.

For example, say you want to get all active workflows. Your curl request will look like this:

```
# For a self-hosted n8n instance
curl -X 'GET' \
  '<N8N_HOST>:<N8N_PORT>/<N8N_PATH>/api/v<version-number>/workflows?
active=true' \
  -H 'accept: application/json' \
  -H 'X-N8N-API-KEY: <your-api-key>'

# For n8n Cloud
curl -X 'GET' \
  '<your-cloud-instance>/api/v<version-number>/workflows?
active=true' \
  -H 'accept: application/json' \
  -H 'X-N8N-API-KEY: <your-api-key>'
```

Delete an API key

1. Log in to n8n.
2. Go to **Settings > n8n API**.
3. Select **Delete** next to the key you want to delete.
4. Confirm the delete by selecting **Delete Forever**.

API pagination

The default page size is 100 results. You can change the page size limit. The maximum permitted size is 250.

When a response contains more than one page, it includes a cursor, which you can use to request the next pages.

For example, say you want to get all active workflows, 150 at a time.

Get the first page:

```
# For a self-hosted n8n instance
curl -X 'GET' \
  '<N8N_HOST>:<N8N_PORT>/<N8N_PATH>/api/v<version-number>/workflows?
active=true&limit=150' \
  -H 'accept: application/json' \
  -H 'X-N8N-API-KEY: <your-api-key>'

# For n8n Cloud
curl -X 'GET' \
  '<your-cloud-instance>/api/v<version-number>/workflows?
active=true&limit=150' \
  -H 'accept: application/json' \
  -H 'X-N8N-API-KEY: <your-api-key>'
```

The response is in JSON format, and includes a `nextCursor` value. This is an example response.

```
{
  "data": [
    // The response contains an object for each workflow
    {
      // Workflow data
    }
  ],
  "nextCursor": "MTIzZTQ1NjctZTg5Yi0xMmQzLWE0NTYtNDI2NjE0MTc0MDA"
}
```

Then to request the next page:

```
# For a self-hosted n8n instance
curl -X 'GET' \
  '<N8N_HOST>:<N8N_PORT>/<N8N_PATH>/api/v<version-number>/workflows?
active=true&limit=150&cursor=MTIzZTQ1NjctZTg5Yi0xMmQzLWE0NTYtNDI2NjE
0MTc0MDA' \
  -H 'accept: application/json'

# For n8n Cloud
curl -X 'GET' \
  '<your-cloud-instance>/api/v<version-number>/workflows?
active=true&limit=150&cursor=MTIzZTQ1NjctZTg5Yi0xMmQzLWE0NTYtNDI2NjE
0MTc0MDA' \
  -H 'accept: application/json'
```

Using an API playground

This documentation site provides a playground to test out calls. Self-hosted users also have access to a built-in playground hosted as part of their instance.

Documentation playground

You can test API calls from this site's [API reference](#). You need to set your server's base URL and instance name, and add an API key.

n8n uses [Scalar's](#) open source API platform to power this functionality.

Built-in playground

The n8n API comes with a built-in Swagger UI playground in self-hosted versions. This provides interactive documentation, where you can try out requests. The path to access the playground depends on your hosting.

n8n constructs the path from values set in your environment variables:

`N8N_HOST:N8N_PORT/N8N_PATH/api/v<api-version-number>/docs`

The API version number is 1. There may be multiple versions available in the future.

The API includes built-in documentation about credential formats. This is available using the `credentials` endpoint:

`N8N_HOST:N8N_PORT/N8N_PATH/api/v<api-version-number>/credentials/schema/{credentialTypeName}`

n8n Embed

n8n Embed is part of n8n's paid offering. Using Embed, you can white label n8n, or incorporate it in your software as part of your commercial product.

For more information about when to use Embed, as well as costs and licensing processes, refer to [Embed](#) on the n8n website.

Support

The [community forum](#) can help with various issues. If you are a current Embed customer, you can also contact n8n support, using the email provided when you bought the license.

Russia and Belarus

n8n Embed isn't available in Russia and Belarus. Refer to n8n's blog post [Update on n8n cloud accounts in Russia and Belarus](#) for more information.

Prerequisites

-8<- “_snippets/embed-license.md”

The requirements provided here are an example based on n8n Cloud and are for illustrative purposes only. Your requirements may vary depending on the number of users, workflows, and executions. Contact n8n for more information.

| Component | Sizing | Supported |
|-----------|--|-----------------------------|
| CPU/vCPU | Minimum 10 CPU cycles, scaling as needed | Any public or private cloud |
| Database | 512 MB - 4 GB SSD | SQLite or PostgreSQL |
| Memory | 320 MB - 2 GB | |

CPU considerations

n8n isn't CPU intensive so even small instances (of providers such as AWS and GCP) should be enough for most use cases. Usually, memory requirements supersede CPU requirements, so focus resources there when planning your infrastructure.

Database considerations

n8n uses its database to store [credentials](#), past executions, and workflows.

A core feature of n8n is the flexibility to choose a database. All the supported databases have different advantages and disadvantages, which you have to consider individually and pick the one that best suits your needs. By default n8n creates an SQLite database if no database exists at the given location.

n8n recommends that every n8n instance have a dedicated database. This helps to prevent dependencies and potential performance degradation. If it isn't possible to provide a dedicated database for every n8n instance, n8n recommends making use of Postgres's schema feature.

For Postgres, the database must already exist on the DB-instance. The database user for the n8n process needs to have full permissions on all tables that they're using or creating. n8n creates and maintains the database schema.

Best practices

- SSD storage.
- In containerized cloud environments, ensure that the volume is persisted and mounted when stopping/starting a container. If not, all data is lost.
- If using Postgres, don't use the `tablePrefix` configuration option. It will be deprecated in the near future.
- Pay attention to the changelog of new versions and consider reverting migrations before downgrading.
- Set up at least the basic database security and stability mechanisms such as IP allow lists and backups.

Memory considerations

An n8n instance doesn't typically require large amounts of available memory. For example an n8n Cloud instance at idle requires ~100MB. It's the nature of your workflows and the data being processed that determines your memory requirements.

For example, while most nodes just pass data to the next node in the workflow, the [Code node](#) creates a pre-processing and post-processing copy of the data. When dealing with large binary files, this can consume all available resources.

Deployment

-8<- “_snippets/embed-license.md”

See the [hosting documentation](#) for detailed setup options.

User data

n8n recommends that you follow the same or similar practices used internally for n8n Cloud: Save user data using [Rook](#) and, if an n8n server goes down, a new instance starts on another machine using the same data.

Due to this, you don't need to use backups except in case of a catastrophic failure, or when a user wants to reactivate their account within your prescribed retention period (two weeks for n8n Cloud).

Backups

n8n recommends creating nightly backups by attaching another container, and copying all data to this second container. In this manner, RAM usage is negligible, and so doesn't impact the amount of users you can place on the server.

Restarting

If your instance is down or restarting, missed executions (for example, Cron or Webhook nodes) during this time aren't recoverable. If it's important for you to maintain 100% uptime, you need to build another proxy in front of it which caches the data.

Configuration

-8<- “_snippets/embed-license.md”

Authentication

You can secure n8n by setting up [User management](#), n8n's built-in authentication feature.

n8n supports [LDAP](#) and [SAML](#).

Credential overwrites

To offer OAuth login to users, it's possible to overwrite [credentials](#) on a global basis. This credential data isn't visible to users but the backend uses it automatically.

In the Editor UI, n8n hides all overwritten fields by default. This means that users are able to authenticate using OAuth by pressing the "connect" button on the credentials.

n8n offers two ways to apply credential overwrites: using Environment Variable and using the REST API.

Using environment variables

You can set credential overwrites using environment variable by setting the `CREDENTIALS_OVERWRITE_DATA` to `{ CREDENTIAL_NAME: { PARAMETER: VALUE } }`.

Using REST APIs

The recommended way is to load the data using a custom REST endpoint. Set the `CREDENTIALS_OVERWRITE_ENDPOINT` to a path under which this endpoint should be made available. You can set `CREDENTIALS_OVERWRITE_ENDPOINT_AUTH_TOKEN` to require a token for accessing the endpoint. When this token is configured, the endpoint is only accessible if the token is included in the Authorization header as a Bearer token.

For example:

1. Activate the endpoint by setting the environment variable in the environment n8n runs under:

```
export CREDENTIALS_OVERWRITE_ENDPOINT=send-credentials
```

2. A JSON file with the credentials to overwrite is then needed. For example, a `oauth-credentials.json` file to overwrite credentials for Asana and GitHub could look like this:

```
{
  "asanaOAuth2Api": {
    "clientId": "<id>",
    "clientSecret": "<secret>"
  },
  "githubOAuth2Api": {
    "clientId": "<id>",
    "clientSecret": "<secret>"
  }
}
```

3. Then apply it to the instance by sending it using curl:

```
curl -H "Content-Type: application/json" --data @oauth-credentials.json http://localhost:5678/send-credentials
```

In case `CREDENTIALS_OVERWRITE_ENDPOINT_AUTH_TOKEN` is set to `secure-token`, the curl command will be:

```
```sh
curl -H "Content-Type: application/json" -H "Authorization: Bearer
secure-token" --data @oauth-credentials.json
http://localhost:5678/send-credentials
```

```

Persistence

To store credential overwrites in the database and propagate them automatically to all workers in multi-instance/queue mode, enable:

```
export CREDENTIALS_OVERWRITE_PERSISTENCE=true
```

When enabled, n8n stores the encrypted overwrites in the `settings` table and broadcasts a `reload-overwrite-credentials` event so that workers reload the latest values. When disabled, overwrites remain in memory on the process that loaded them and aren't propagated to workers or preserved across restarts.

Environment variables

n8n has many environment variables you can configure. Here are the most relevant environment variables for your hosted solution:

| Variable | Type | Default | Desc |
|---------------------------------|------------------|---------|--|
| EXECUTIONS_TIMEOUT | Number | -1 | Sets the timeout for individual executions (in seconds). If set to -1, it will be up to the worker to decide when to stop the execution. |
| EXECUTIONS_DATA_PRUNE | Boolean | true | Whether to prune old data after a roll-in. This is useful for clearing out old data when a node is deleted. |
| EXECUTIONS_DATA_MAX_AGE | Number | 336 | The maximum age of data in hours before it is deleted. |
| EXECUTIONS_DATA_PRUNE_MAX_COUNT | Number | 10000 | Maximum number of executions to delete if the data limit is reached. |
| NODES_EXCLUDE | Array of strings | - | Specifies nodes to exclude from the execution queue. It can be a single node name or a list of node names enclosed in brackets. For example, <code>["node1", "node2"]</code> . |

| | | | |
|-----------------------|---------------------------|--|---|
| NODES_INCLUDE | Array of strings | -
base.
\n8n
base. | Spec
load. |
| N8N_TEMPLATES_ENABLED | Boolean true | Enab
<u>temp</u>
disab | Chan
your
temp
that t
work
libra |
| N8N_TEMPLATES_HOST | String https://api.n8n.io | provi
endp
respo
n8n's
<u>Work</u>
more | Chan
your
temp
that t
work
libra |

Backend hooks

It's possible to define external hooks that n8n executes whenever a specific operation runs. You can use these, for example, to log data, change data, or forbid an action by throwing an error.

Available hooks

| Hook | Arguments | Description |
|--------------------|---------------------------------------|---|
| credentials.create | [credentialData:
ICredentialsDb] | Called before new credentials get created. Use to restrict the number of credentials. |
| credentials.delete | [id: credentialId] | Called before credentials get deleted. |
| credentials.update | [credentialData:
ICredentialsDb] | Called before existing credentials are saved. |
| frontend.settings | [frontendSettings:
IN8nUISettings] | Gets called on n8n startup. Allows you to, for example, overwrite frontend data like the displayed OAuth URL. |
| n8n.ready | [app: App] | Called once n8n is ready. Use to, for example, register |

| | | |
|----------------------|---|---|
| n8n.stop | | custom API endpoints. |
| oauth1.authenticate | [oAuthOptions:
clientOAuth1.Options,
oauthrequestData:
{oauth_callback:
string}] | Called when an n8n process gets stopped. Allows you to save some process data. |
| oauth2.callback | [oAuth2Parameters:
{clientId: string,
clientSecret: string
\ undefined,
accessTokenUri:
string,
authorizationUri:
string, redirectUri:
string, scopes:
string[]}] | Called before an OAuth1 authentication. Use to overwrite an OAuth callback URL. |
| workflow.activate | [workflowData:
IWorkflowDb] | Called in an OAuth2 callback. Use to overwrite an OAuth callback URL. |
| workflow.afterCreate | [workflowId: string] | Called before a workflow gets activated. Use to restrict the number of active workflows. |
| workflow.afterDelete | [workflowId: string] | Called after a workflow gets created. |
| workflow.afterUpdate | [workflowData:
IWorkflowBase] | Called after a workflow gets deleted. |
| workflow.create | [workflowData:
IWorkflowBase] | Called after an existing workflow gets saved. |
| workflow.delete | [workflowId: string] | Called before a workflow gets saved. |
| workflow.postExecute | [run: IRun,
workflowData:
IWorkflowBase] | Called before a workflow gets deleted. |
| workflow.preExecute | [workflow: Workflow:
mode:
WorkflowExecuteMode] | Called after a workflow gets executed. |
| | | Called before a workflow gets executed. Allows you to count or limit the number of workflow executions. |

| | | |
|--------------------------------------|--|---|
| <code>workflow.update</code> | <code>[workflowData: IWorkflowBase]</code> | Called before an existing workflow gets saved. |
| <code>workflow.afterArchive</code> | <code>[workflowId: string]</code> | Called after you archive a workflow. |
| <code>workflow.afterUnarchive</code> | <code>[workflowId: string]</code> | Called after you restore a workflow from the archive. |

Registering hooks

Set hooks by registering a hook file that contains the hook functions. To register a hook, set the environment variable `EXTERNAL_HOOK_FILES`.

You can set the variable to a single file:

```
EXTERNAL_HOOK_FILES=/data/hook.js
```

Or to contain multiple files separated by a colon:

```
EXTERNAL_HOOK_FILES=/data/hook1.js:/data/hook2.js
```

Backend hook files

Hook files are regular JavaScript files that have the following format:

```
module.exports = {
  "frontend": {
    "settings": [
      async function (settings) {
        settings.oauthCallbackUrls.oauth1 =
        'https://n8n.example.com/oauth1/callback';
        settings.oauthCallbackUrls.oauth2 =
        'https://n8n.example.com/oauth2/callback';
      }
    ],
    "workflow": {
      "activate": [
        async function (workflowData) {
          const activeWorkflows = await
this.dbCollections.Workflow.count({ active: true });

          if (activeWorkflows > 1) {
            throw new Error(
              'Active workflow limit reached.'
            );
          }
        }
      ]
    }
  }
}
```

Backend hook functions

A hook or a hook file can contain multiple hook functions, with all functions executed one after another.

If the parameters of the hook function are objects, it's possible to change the data of that parameter to change the behavior of n8n.

You can also access the database in any hook function using `this.dbCollections` (refer to the code sample in [Backend hook files](#)).

Frontend external hooks

Like backend external hooks, it's possible to define external hooks in the frontend code that get executed by n8n whenever a user performs a specific operation. You can use them, for example, to log data and change data.

Available hooks

| Hook | Description |
|---|---|
| <code>credentialsEdit.credentialTypeChanged</code> | Called when an existing credential's type changes. |
| <code>credentials.create</code> | Called when someone creates a new credential. |
| <code>credentialsList.dialogVisibleChanged</code> | |
| <code>dataDisplay.nodeTypeChanged</code> | |
| <code>dataDisplay.onDocumentationUrlClick</code> | Called when someone selects the help documentation link. |
| <code>execution.open</code> | Called when an existing execution opens. |
| <code>executionsList.openDialog</code> | Called when someone selects an execution from existing Workflow Executions. |
| <code>expressionEdit.itemSelected</code> | |
| <code>expressionEdit.dialogVisibleChanged</code> | |
| <code>nodeCreateList.filteredNodeTypesComputed</code> | |
| <code>nodeCreateList.nodeFilterChanged</code> | Called when someone makes any changes to the node panel filter. |
| <code>nodeCreateList.selectedTypeChanged</code> | |
| <code>nodeCreateList.mounted</code> | |
| <code>nodeCreateList.destroyed</code> | |
| <code>nodeSettings.credentialSelected</code> | |
| <code>nodeSettings.valueChanged</code> | |
| <code>nodeView.createNodeActiveChanged</code> | |
| <code>nodeView.addNodeButton</code> | |
| <code>nodeView.createNodeActiveChanged</code> | |
| <code>nodeView.mount</code> | |
| <code>pushConnection.executionFinished</code> | |
| <code>showMessage.showError</code> | |
| <code>runData.displayModeChanged</code> | |
| <code>workflow.activeChange</code> | |

| | |
|--|---|
| <code>workflow.activeChangeCurrent</code> | |
| <code>workflow.afterUpdate</code> | Called when someone updates an existing workflow. |
| <code>workflow.open</code> | |
| <code>workflowRun.runError</code> | |
| <code>workflowRun.runWorkflow</code> | Called when a workflow executes. |
| <code>workflowSettings.dialogVisibleChanged</code> | |
| <code>workflowSettings.saveSettings</code> | Called when someone saves the settings of a workflow. |

Registering hooks

You can set hooks by loading the hooks script on the page. One way to do this is by creating a hooks file in the project and adding a script tag in your `editor-ui/public/index.html` file:

Frontend hook files

Frontend external hook files are regular JavaScript files which have the following format:

```
window.n8nExternalHooks = {
    nodeView: [
        mount: [
            function (store, meta) {
                // do something
            },
        ],
        createNodeActiveChanged: [
            function (store, meta) {
                // do something
            },
            function (store, meta) {
                // do something else
            },
        ],
        addNodeButton: [
            function (store, meta) {
                // do something
            },
        ],
    ],
};
```

Frontend hook functions

You can define multiple hook functions per hook. Each hook function is invoked with the following arguments arguments:

- `store`: The Vuex store object. You can use this to change or get data from the store.
- `metadata`: The object that contains any data provided by the hook. To see what's passed, search for the hook in the `editor-ui` package.

Workflow management in Embed

-8<- “_snippets/embed-license.md”

When managing an embedded n8n deployment, spanning across teams or organizations, you will likely need to run the same (or similar) workflows for multiple users. There are two available options for doing so:

| Solution | Pros | Cons |
|---|--|--|
| Create a workflow for each user | No limitation on how workflow starts (can use any trigger) | Requires managing multiple workflows. |
| Create a single workflow, and pass it user credentials when executing | Simplified workflow management (only need to change one workflow). | To run the workflow, your product must call it |

Workflow per user

There are three general steps to follow:

- Obtain the credentials for each user, and any additional parameters that may be required based on the workflow.
- Create the n8n credentials for this user.
- Create the workflow.

1. Obtain user credentials

Here you need to capture all credentials for any node/service this user must authenticate with, along with any additional parameters required for the particular workflow. The credentials and any parameters needed will depend on your workflow and what you are trying to do.

2. Create user credentials

After all relevant credential details have been obtained, you can proceed to create the relevant service credentials in n8n. This can be done using the Editor UI or API call.

Using the Editor UI

1. From the menu select **Credentials > New**.
2. Use the drop-down to select the **Credential type** to create, for example *Airtable*. [Image: Create New Credentials drop-down]
3. In the **Create New Credentials** modal, enter the corresponding credentials details for the user, and select the nodes that will have access to these credentials. [Image: Create New Credentials modal]

4. Click **Create** to finish and save.

Using the API

The frontend API used by the Editor UI can also be called to achieve the same result. The API endpoint is in the format: `https://<n8n-domain>/rest/credentials`.

For example, to create the credentials in the Editor UI example above, the request would be:

```
POST https://<n8n-domain>/rest/credentials
```

With the request body:

```
{
  "name": "MyAirtable",
  "type": "airtableApi",
  "nodesAccess": [
    {
      "nodeType": "n8n-nodes-base.airtable"
    }
  ],
  "data": {
    "apiKey": "q12we34r5t67yu"
  }
}
```

The response will contain the ID of the new credentials, which you will use when creating the workflow for this user:

```
{
  "data": {
    "name": "MyAirtable",
    "type": "airtableApi",
    "data": {
      "apiKey": "q12we34r5t67yu"
    },
    "nodesAccess": [
      {
        "nodeType": "n8n-nodes-base.airtable",
        "date": "2021-09-10T07:41:27.770Z"
      }
    ],
    "id": "29",
    "createdAt": "2021-09-10T07:41:27.777Z",
    "updatedAt": "2021-09-10T07:41:27.777Z"
  }
}
```

3. Create the workflow

Best practice is to have a “base” workflow that you then duplicate and customize for each new user with their credentials (and any other details).

You can duplicate and customize your template workflow using either the Editor UI or API call.

Using the Editor UI

1. From the menu select **Workflows > Open** to open the template workflow to be duplicated.
2. Select **Workflows > Duplicate**, then enter a name for this new workflow and click **Save**. [Image: Duplicate workflow]
3. Update all relevant nodes to use the credentials for this user (created above).
4. **Save** this workflow set it to **Active** using the toggle in the top-right corner.

Using the API

1. Fetch the JSON of the template workflow using the endpoint:
https://<n8n-domain>/rest/workflows/<workflow_id>

```
GET https://<n8n-domain>/rest/workflows/1012
```

The response will contain the JSON data of the selected workflow:

```
{
  "data": {
    "id": "1012",
    "name": "Nathan's Workflow",
    "active": false,
    "nodes": [
      {
        "parameters": {},
        "name": "Start",
        "type": "n8n-nodes-base.start",
        "typeVersion": 1,
        "position": [
          130,
          640
        ]
      },
      {
        "parameters": {
          "authentication": "headerAuth",
          "url": "https://internal.users.n8n.cloud/webhook/custom-erp",
          "options": {
            "splitIntoItems": true
          },
          "headerParametersUi": {
            "parameter": [
              {
                "name": "unique_id",
                "value": "recLhLYQbzNSFtHNq"
              }
            ]
          }
        },
        "name": "HTTP Request",
        "type": "n8n-nodes-base.httpRequest",
        "typeVersion": 1,
        "position": [
          430,
          300
        ],
        "connections": [
          {
            "id": "1012_start_1012_httpRequest_1",
            "order": 1,
            "x": 130,
            "y": 640,
            "x2": 430,
            "y2": 300
          }
        ]
      }
    ],
    "connections": [
      {
        "id": "1012_start_1012_httpRequest_1",
        "order": 1,
        "x": 130,
        "y": 640,
        "x2": 430,
        "y2": 300
      }
    ]
  }
}
```

```

    "credentials": {
      "httpHeaderAuth": "beginner_course"
    }
  },
  {
    "parameters": {
      "operation": "append",
      "application": "appKBGQfbm6NfW6bv",
      "table": "processingOrders",
      "options": {}
    },
    "name": "Airtable",
    "type": "n8n-nodes-base.airtable",
    "typeVersion": 1,
    "position": [
      990,
      210
    ],
    "credentials": {
      "airtableApi": "Airtable"
    }
  },
  {
    "parameters": {
      "conditions": {
        "string": [
          {
            "value1": "{$json[\"orderStatus\"]}",
            "value2": "processing"
          }
        ]
      }
    },
    "name": "IF",
    "type": "n8n-nodes-base.if",
    "typeVersion": 1,
    "position": [
      630,
      300
    ]
  },
  {
    "parameters": {
      "keepOnlySet": true,
      "values": {
        "number": [
          {
            "name": "=orderId",
            "value": "{$json[\"orderID\"]}"
          }
        ],
        "string": [
          {
            "name": "employeeName",
            "value": "{$json[\"employeeName\"]}"
          }
        ],
        "options": {}
      }
    }
  }
}

```

```

        "name": "Set",
        "type": "n8n-nodes-base.set",
        "typeVersion": 1,
        "position": [
            800,
            210
        ],
    },
    {
        "parameters": {
            "functionCode": "let totalBooked = items.length;\nlet bookedSum = 0;\n\nfor(let i=0; i < items.length; i++) {\n    bookedSum = bookedSum + items[i].json.orderPrice;\n}\n\nreturn [{json: {totalBooked, bookedSum}}]\n"
        },
        "name": "Function",
        "type": "n8n-nodes-base.function",
        "typeVersion": 1,
        "position": [
            800,
            400
        ],
    },
    {
        "parameters": {
            "webhookUri": "https://discord.com/api/webhooks/865213348202151968/oD5_WPDQwtr22Vjd_82QP3-_4b_lGhAeM7RynQ8Js5DzyXrQEnj0zeAQIA6fkilJLxE",
            "text": "=This week we have {{$json[\"totalBooked\"]}} booked orders with a total value of {{$json[\"bookedSum\"]}}. My Unique ID: {{$node[\"HTTP Request\"]}.parameter[\"headerParametersUi\"][\"parameter\"][@] [\"value\"]}}"
        },
        "name": "Discord",
        "type": "n8n-nodes-base.discord",
        "typeVersion": 1,
        "position": [
            1000,
            400
        ],
    },
    {
        "parameters": {
            "triggerTimes": {
                "item": [
                    {
                        "mode": "everyWeek",
                        "hour": 9
                    }
                ]
            }
        },
        "name": "Cron",
        "type": "n8n-nodes-base.cron",
        "typeVersion": 1,
        "position": [
            220,
            300
        ],
    }
]

```

```
        }
    ],
"connections": {
    "HTTP Request": {
        "main": [
            [
                {
                    "node": "IF",
                    "type": "main",
                    "index": 0
                }
            ]
        ],
    },
    "Start": {
        "main": [
            []
        ],
    },
    "IF": {
        "main": [
            [
                [
                    {
                        "node": "Set",
                        "type": "main",
                        "index": 0
                    }
                ],
                [
                    [
                        {
                            "node": "Function",
                            "type": "main",
                            "index": 0
                        }
                    ]
                ]
            ],
    },
    "Set": {
        "main": [
            [
                [
                    {
                        "node": "Airtable",
                        "type": "main",
                        "index": 0
                    }
                ]
            ],
    },
    "Function": {
        "main": [
            [
                [
                    {
                        "node": "Discord",
                        "type": "main",
                        "index": 0
                    }
                ]
            ],
    },
    "Cron": {
```

```

    "main": [
      [
        {
          "node": "HTTP Request",
          "type": "main",
          "index": 0
        }
      ]
    ],
    "createdAt": "2021-07-16T11:15:46.066Z",
    "updatedAt": "2021-07-16T12:05:44.045Z",
    "settings": {},
    "staticData": null,
    "tags": []
  }
}

```

1. Save the returned JSON data and update any relevant credentials and fields for the new user.
2. Create a new workflow using the updated JSON as the request body at endpoint: `https://<n8n-domain>/rest/workflows`

`POST https://<n8n-domain>/rest/workflows/`

The response will contain the ID of the new workflow, which you will use in the next step.

1. Lastly, publish the new workflow:

`PATCH https://<n8n-domain>/rest/workflows/1012`

Passing the additional value `active` in your JSON payload:

```

// ...
"active":true,
"settings": {},
"staticData": null,
"tags": []

```

Single workflow

There are four steps to follow to implement this method:

- Obtain the credentials for each user, and any additional parameters that may be required based on the workflow. See [Obtain user credentials](#) above.
- Create the n8n credentials for this user. See [Create user credentials](#) above.
- Create the workflow.
- Call the workflow as needed.

Create the workflow

The details and scope of this workflow will vary greatly according to the individual use case, however there are a few design implementations to keep in mind:

- This workflow must be triggered by a [Webhook](#) node.
- The incoming webhook call must contain the user's credentials and any other workflow parameters required.
- Each node where the user's credentials are needed should use an [expression](#) so that the node's credential field reads the credential provided in the webhook call.
- Save and publish the workflow, ensuring the production URL is selected for the Webhook node. Refer to [webhook node](#) for more information.

Call the workflow

For each new user, or for any existing user as may be needed, call the webhook defined as the workflow trigger and provide the necessary credentials (and any other workflow parameters).

Workflow templates

-8<- “_snippets/embed-license.md”

n8n provides a library of workflow [templates](#). When embedding n8n, you can:

- Continue to use n8n's workflow templates library (this is the default behavior)
- Disable workflow templates
- Create your own workflow templates library

Disable workflow templates

-8<- “_snippets/workflows/templates/disable-templates.md”

Use your own workflow templates library

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

Add your workflows to the n8n library

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

White labelling

-8<- “_snippets/embed-license.md”

White labelling n8n means customizing the frontend styling and assets to match your brand identity. The process involves changing two packages in n8n's source code github.com/n8n-io/n8n:

- [packages/frontend/@n8n/design-system](#): n8n's [storybook](#) design system with CSS styles and Vue.js components
- [packages/frontend/editor-ui](#): n8n's [Vue.js](#) frontend build with

Vite.js

Prerequisites

You need the following installed on your development machine:

-8<- “_snippets/integrations/creating-nodes/prerequisites.md”

Create a fork of [n8n’s repository](#) and clone your new repository.

```
git clone https://github.com/<your-organization>/n8n.git n8n
cd n8n
```

Install all dependencies, build and start n8n.

```
npm install
npm run build
npm run start
```

Whenever you make changes you need to rebuild and restart n8n.
While developing you can use `npm run dev` to automatically rebuild and restart n8n anytime you make code changes.

Theme colors

To customize theme colors open [packages/frontend/@n8n/design-system](#) and start with:

- [packages/frontend/@n8n/design-system/src/css/_tokens.scss](#)
- [packages/frontend/@n8n/design-system/src/css/_tokens.dark.scss](#)

At the top of `_tokens.scss` you will find `--color-primary` variables as HSL colors:

```
@mixin theme {
  --color-primary-h: 6.9;
  --color-primary-s: 100%;
  --color-primary-l: 67.6%;
```

In the following example the primary color changes to `#0099ff`. To convert to HSL you can use a [color converter tool](#).

```
@mixin theme {
  --color-primary-h: 204;
  --color-primary-s: 100%;
  --color-primary-l: 50%;
```

[Image: Example Theme Color Customization]

Theme logos

To change the editor’s logo assets look into [packages/frontend/editor-ui/public](#) and replace:

- favicon-16x16.png
- favicon-32x32.png
- favicon.ico
- n8n-logo.svg

- n8n-logo-collapsed.svg
- n8n-logo-expanded.svg

Replace these logo assets. n8n uses them in Vue.js components, including:

- [MainSidebar.vue](#): top/left logo in the main sidebar.
- [Logo.vue](#): reused in other components.

In the following example replace n8n-logo-collapsed.svg and n8n-logo-expanded.svg to update the main sidebar's logo assets.

[Image: Example Logo Main Sidebar]

If your logo assets require different sizing or placement you can customize SCSS styles at the bottom of [MainSidebar.vue](#).

```
.logoItem {
  display: flex;
  justify-content: space-between;
  height: $header-height;
  line-height: $header-height;
  margin: 0 !important;
  border-radius: 0 !important;
  border-bottom: var(--border-width-base) var(--border-style-base)
  var(--color-background-xlight);
  cursor: default;

  &:hover, &:global(.is-active):hover {
    background-color: initial !important;
  }

  * { vertical-align: middle; }
  .icon {
    height: 18px;
    position: relative;
    left: 6px;
  }
}
```

Text localization

To change all text occurrences like n8n or n8n.io to your brand identity you can customize n8n's English internationalization file: [packages/frontend/@n8n/i18n/src/locales/en.json](#).

n8n uses the [Vue I18n](#) internationalization plugin for Vue.js to translate the majority of UI texts. To search and replace text occurrences inside en.json you can use [Linked locale messages](#).

In the following example add the _brand.name translation key to white label n8n's [AboutModal.vue](#).

```
{
  "_brand.name": "My Brand",
  //replace n8n with link to _brand.name
  "about.aboutN8n": "About @:_brand.name",
  "about.n8nVersion": "@:_brand.name Version",
}
```

[Image: Example About Modal Localization]

Window title

To change n8n's window title to your brand name, edit the following:

- [packages/frontend/editor-ui/index.html](#)
- [packages/frontend/editor-ui/src/composables/useDocumentTitle.ts](#)

The following example replaces all occurrences of n8n and n8n.io with My Brand in index.html and useDocumentTitle.ts.

```
<!DOCTYPE html>
<html lang="en">
<head>

    <title>My Brand - Workflow Automation</title>
</head>

import { useSettingsStore } from '@/stores/settings.store';

// replace n8n
const DEFAULT_TITLE = 'My Brand';
const DEFAULT_TAGLINE = 'Workflow Automation';
```

[Image: Example Window Title Localization]