

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.
 - **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",
      "name2": "value2",
      "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
Add Blank To Links	Whether to open links a new window (enabled) or not (disabled).	Disabled
Automatic Linking To URLs	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
Backslash Escapes HTML Tags	Whether to allow backslash escaping of HTML tags (enabled) or not (disabled). When enabled, n8n escapes any < or > prefaced with \. For example, \<div> renders as <div>.	Disabled
Complete HTML Document	Whether to output a complete HTML document (enabled) or an HTML fragment (disabled). A complete HTML document includes the <DOCTYPE HTML> declaration,	Disabled

	<html> and <body> tags, and the <head> element.	
Customized Header ID	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
Emoji Support	Whether to support emojis (enabled) or not (disabled).	Disabled.
Encode Emails	Whether to transform ASCII character emails into their equivalent decimal entities (enabled) or not (disabled).	Enabled
Exclude Trailing Punctuation From URLs	Whether to exclude trailing punctuation from automatically linked URLs (enabled) or not (disabled). For use with Automatic Linking To URLs .	Disabled
GitHub Code Blocks	Whether to enable GitHub Flavored Markdown code blocks (enabled) or not (disabled).	Enabled
GitHub Compatible Header IDs	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
GitHub Mention Link	Change the link used with GitHub Mentions .	Disabled
GitHub Mentions	Whether to support tagging GitHub users with @ (enabled) or not (disabled). When enabled, n8n replaces @name with https://github.com/name.	Disabled
GitHub Task Lists	Whether to support GitHub Flavored Markdown task lists (enabled) or not (disabled).	Disabled
Header Level Start	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
Mandatory Space Before Header	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	

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

Support (enabled) or not (disabled).

HTML to Markdown options

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

		<pre> (disabled). Disabled **Strong Delimiter** Specify the characters for`. </pre>
Style For Code Block	Specify the styling for code blocks. Options are Fence and Indented .	Fence
Text Replacement Pattern	Define a text replacement pattern using regex.	None
Treat As Blocks	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

[
  {
    "fieldName": "Date Field",
    "fieldType": "date",
    "formatDate": "mm/dd/yyyy", // how to format received date in
n8n
    "requiredField": true
  },
  {
    "fieldName": "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
  "acceptFileTypes": ".jpg, .png" // allowed file types
},
{
  "fieldLabel": "Number",
  "fieldType": "number"
},
{
  "fieldLabel": "Password",
  "fieldType": "password"
},
{
  // "fieldType": "text" can be omitted since it's the default
  "fieldLabel": "Text"
},
{
  "fieldLabel": "Textarea",
  "fieldType": "textarea",
  "defaultValue": "Lorem ipsum."
},
{
  "fieldType": "html",
  "elementName": "content", // Optional field. It can be used to
  "html": "<div>Custom element</div>"
},
{
  "fieldLabel": "Checkboxes",
  "fieldType": "checkbox",
  "fieldOptions": {
    "values": [
      {
        "option": "option 1"
      },
      {

```



```

        "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
 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
 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 `
` 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
 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 (
) 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 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:

id	name	job	last_updated
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:

id	name	job	last_updated
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:

id	name	job	last_updated
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:

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

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

id	name	job	last_updated
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:

id	name	job	last_updated
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):

id	name	job	last_updated
----	------	-----	--------------

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:

id	name	job	last_updated
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:

id	name	job	last_updated
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

- Open the Code node and uncomment (remove the `//` from) the lines for “Harry Nilsson” and “Kylie Minogue.”
- 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):

id	name	job	last_updated
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:

id	name	job	last_updated
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 Output 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](https://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@sample.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@sample.com>, for example: Nathan Doe <nate@n8n.io>. Use a comma to separate multiple email addresses: first@sample.com, "Name" <second@sample.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 Listen for test event 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 Airtable 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 templating 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:** Delete 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`

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`.
- **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`.
- **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 a contact
 - Retrieve a 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-clfd-414d-85c8-84f33e92618a",
  "_xform_id_string": "ajXVJpBkTD5tB4Nu9QXpgm",
  "_uuid": "2356cbbe-clfd-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 to 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. A 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 }}`. 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 }}` 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 **Analyze 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", helloworldwithoutquotes, "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 n&n 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

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

SIGNAL4 node

Use the SIGNAL4 node to automate work in SIGNAL4, and integrate SIGNAL4 with other applications. n8n supports sending and resolving alerts with SIGNAL4.

On this page, you'll find a list of operations the SIGNAL4 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	<u>conversations.create</u>
Channel	Get	<u>conversations.info</u>
Channel	Get Many	<u>conversations.list</u>
Channel	History	<u>conversations.history</u>
Channel	Invite	<u>conversations.invite</u>
Channel	Join	<u>conversations.join</u>
Channel	Kick	<u>conversations.kick</u>
Channel	Leave	<u>conversations.leave</u>
Channel	Member	<u>conversations.members</u>
Channel	Open	<u>conversations.open</u>
Channel	Rename	<u>conversations.rename</u>
Channel	Replies	<u>conversations.replies</u>
Channel	Set Purpose	<u>conversations.setPurpose</u>
Channel	Set Topic	<u>conversations.setTopic</u>
Channel	Unarchive	<u>conversations.unarchive</u>
File	Get	<u>files.info</u>
File	Get Many	<u>files.list</u>
File	Upload	<u>files.upload</u>
Message	Delete	<u>chat.delete</u>
Message	Get Permalink	<u>chat.getPermalink</u>
Message	Search	<u>search.messages</u>
Message	Send	<u>chat.postMessage</u>
Message	Send and Wait for Approval	<u>chat.postMessage</u>
Message	Update	<u>chat.update</u>
Reaction	Add	<u>reactions.add</u>
Reaction	Get	<u>reactions.get</u>
Reaction	Remove	<u>reactions.remove</u>
Star	Add	<u>stars.add</u>
Star	Delete	<u>stars.remove</u>
Star	Get Many	<u>stars.list</u>
User	Get	<u>users.info</u>
User	Get Many	<u>users.list</u>
User	Get User's Profile	<u>users.profile.get</u>
User	Get User's Status	<u>users.getPresence</u>
User	Update User's Profile	<u>users.profile.set</u>
User Group	Create	<u>usergroups.create</u>
User Group	Disable	<u>usergroups.disable</u>
User Group	Enable	<u>usergroups.enable</u>

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 Alpha2 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

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

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 by many 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 produces 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"},  
      "name": "webhook", "endpoint": "TESTING_WEBHOOK_NAME" }  
    },  
    {  
      "trigger": { "name": "AnyEvent"},  
      "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": [
    {
      "delegate": false,
      "trigger": { "name": "AnyEvent"},
      "notifier": {
        "name": "webhook", "endpoint": "TESTING_WEBHOOK_NAME" }
    },
    {
      "delegate": false,
      "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](#)
- [Grog 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](#) * [Wolfram|Alpha](#) * [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 Code node .	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

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": "hns",
            "hnsParameters": {
                "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 summarize 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 summarize 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 summarize 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 summarize 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 summarize 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 summarize 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 summarize 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 summarize 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 [NO_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/app-settings`, `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 value 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