Respond to Webhook

Use the Respond to Webhook node to control the response to incoming webhooks. This node works with the Webhook node.

Runs once for the first data item

The Respond to Webhook node runs once, using the first incoming data item. Refer to Return more than one data item for more information.

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.

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

Creating an API endpoint
by Jonathan

View template details

Create a Branded AI-Powered Website Chatbot
by Wayne Simpson

View template details

⚡ AI-Powered YouTube Video Summarization & Analysis
by Joseph LePage

View template details

Browse Respond to Webhook integration templates, or search all templates

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)

> **Deprecated in 1.22.0**
>
> n8n 1.22.0 added support for returning all data items using the **All Incoming Items** option. n8n recommends upgrading to the latest version of n8n, instead of using the workarounds described in this section.

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)**.