Node Guide: GET API

Overview

The **GET API** node allows you to **call any external GET endpoint** and fetch data from it. You can configure the endpoint URL, include headers, query parameters, path variables, and authentication options. The response can be saved in variables for use in the rest of your workflow.

This is useful when your workflow needs to pull data from external systems like CRMs, ticketing tools, databases, or any third-party APIs.

What It Does

Once triggered:

- Sends a **GET request** to the configured API URL

- Passes along any required parameters, headers, or path variables

- Handles authentication if needed

- Stores the **response code**, **response body**, and **headers** in variables

Configuration Details

1. **GET Endpoint URL**

   - Enter the URL of the API you want to call

   - You can include dynamic path variables like:
     `https://api.example.com/user/{{user_id}}`

2. **Query Parameters & Headers**

   - Add key-value pairs for any request parameters or custom headers

○ Values can be typed manually or passed through variables

3. **Authentication**

   ○ Choose from **Basic**, **Bearer**, or **OAuth2**

   ○ Set the credentials directly or via variables

4. **Test Before Apply**

   ○ You can test the API call within the configuration screen

   ○ This helps ensure the request is valid and the response is as expected

---

Inputs

- **GET Endpoint URL**

- **Headers, Params, Path Variables (Optional)**

- **Authentication Details (Optional)**

Outputs

- `response_code` — The status code returned by the API

- `response_body` — The actual response data (usually in JSON)

- `response_headers` — Metadata from the API response

All outputs can be stored in variables for use in following nodes.

When to Use

Use the GET API node when you want to:

- Fetch user, ticket, or product information from a service

- Query analytics or log data

- Pull live information into your workflow dynamically

Example Flow: Check User Status from CRM

Scenario

You want to verify if a user is active in your CRM system based on their user ID.

Flow Steps

1. **GET API Node**

   - URL: `https://crm.example.com/api/user/{{user_id}}`

   - Authentication: Bearer Token from variable

   - Output: Save response_body to `user_data` variable

2. **Condition Node**

   - Check if `user_data.status` equals `"active"`

   - If yes, continue the workflow; if not, send a different message

---

Summary of the Flow

- Makes a GET request to an external system

- Reads and stores the returned information

- Uses that information to guide the workflow