Node Guide: Script

Overview

The Script node allows you to execute custom code within your workflow using supported programming languages. This is useful for logic, transformation, validation, or generating data on the fly.

What This Node Does

- Executes a code snippet written in a selected language
- Can access existing variables from the flow using `{{variable_name}}` syntax
- Returns the output of the script, which can be stored in a variable and used later in the workflow

---

Configuration Details

1. Language Selection

Choose one of the supported script languages from the dropdown:

- JavaScript
- Java
- Python
- C++

2. Script Editor

- Write the code directly in the editor provided
- You can click on prebuilt **standard functions**, **modules**, and **supported built-ins** to insert them into your code

- Use `{{variable_name}}` to insert workflow variable values into your code

3. Output Handling

- Choose whether to store the result of the script in a variable
- If selected, the result will be accessible to downstream nodes via that variable

---

Inputs

- Script: Type directly in the editor
- Variables: Use variables from the dropdown list to insert into your code using `{{variable_name}}`

Outputs

- Result of the script execution
- Captured into a variable if specified

When to Use

Use this node when you want to:

- Perform dynamic calculations
- Apply custom logic, validations, or formatting
- Generate or transform data during the workflow
- Use external libraries or built-in functions to derive results

---

Example Flow: Generate Unique File Name

Scenario

You want to create a filename using a user's ID and the current timestamp.

Flow Steps

1. **Script Node**
    - Language: JavaScript

Script:

javascript
CopyEdit

```
const userId = `{{user_id}}`;
const timestamp = Date.now();
return `${userId}_${timestamp}`;
```

    -
    - Output Variable: `generated_filename`

2. **Upload Node**
    - Uses `generated_filename` as the destination file name

Summary

The Script node is a flexible execution block where you can run JavaScript, Python, Java, or C++ code inside your workflow. It supports variable injection, standard libraries, and lets you optionally return output for further use.