

Node Guide: Agent Executor

Overview

The **Agent Executor** node allows you to execute an existing agent from another application or workflow within your current flow. This promotes reusability and avoids duplicating logic across different agents.

What This Node Does

- Executes a selected agent from the list of available agents
 - Passes the required input variables to that agent
 - Captures the output of the executed agent into a variable for further use in the current flow
-

Configuration Details

1. Agent Selection

- Choose the agent you want to execute from the available dropdown list
- The agent must be published or accessible from another application or part of the platform

2. Input Mapping

- If the selected agent requires inputs, map them using existing variables from the dropdown.

3. Output Storage

- Capture the result/output of the executed agent into a variable
 - This output can then be used in subsequent nodes within your current flow
-

Inputs

- Agent: Select from the dropdown list of agents
 - Inputs to Agent: Pass using mapped variables
-

Outputs

- Output of the selected agent
 - Can be stored in a variable and used downstream
-

When to Use

Use this node when you want to:

- Reuse logic from agents built in other applications
 - Reduce redundancy by referencing already configured flows
 - Modularize complex logic across agents and reuse them like subroutines
 - Chain multiple agents together for more powerful workflows
-

Example Flow: Reuse a Document Analyzer Agent

Scenario

You have an existing agent that processes documents and returns key metadata. Instead of rebuilding it, you want to use it in a new insurance claims agent.

Flow Steps

1. Agent Executor Node

- Agent: **Document Analyzer**
- Input: Pass uploaded document ID from variable `{{uploaded_doc_id}}`

- Output Variable: `analyzed_data`

2. Send Text Node

- Message: "The key details extracted are: {{analyzed_data}}"

Summary

The Agent Executor node is ideal for reusing agent logic across applications. It supports modular, maintainable workflows by allowing one agent to call and use another without duplicating the underlying logic.

Let me know if you'd like visual diagrams or examples of multi-agent chaining!