ReAct AI Agent node documentation

# 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.

No memory

The ReAct agent doesn't support memory sub-nodes. This means it can't recall previous prompts or simulate an ongoing conversation.

## Node parameters#

Configure the ReAct Agent using the following parameters.

### Prompt#

Select how you want the node to construct the prompt (also known as the user's query or input from the chat).

Choose from:

• Take from previous node automatically: If you select this option, the node expects an input from a previous node called chatInput.

chatInput

• Define below: If you select this option, provide either static text or an expression for dynamic content to serve as the prompt in the Prompt (User Message) field.

### Require Specific Output Format#

This parameter controls whether you want the node to require a specific output format. When turned on, n8n prompts you to connect one of these output parsers to the node:

• Auto-fixing Output Parser

• Item List Output Parser

• Structured Output Parser

## 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.

{input}

• {agent\_scratchpad}: Information to remember for the next iteration.

{agent\_scratchpad}

### 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#

Select whether to include intermediate steps the agent took in the final output (turned on) or not (turned off).

This could be useful for further refining the agent's behavior based on the steps it took.

## 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.

## AI glossary#

• completion: Completions are the responses generated by a model like GPT.

• hallucinations: Hallucination in AI is when an LLM (large language model) mistakenly perceives patterns or objects that don't exist.

• vector database: A vector database stores mathematical representations of information. Use with embeddings and retrievers to create a database that your AI can access when answering questions.

• vector store: A vector store, or vector database, stores mathematical representations of information. Use with embeddings and retrievers to create a database that your AI can access when answering questions.