Plan and Execute AI Agent node documentation

# Plan and Execute Agent node#

The Plan and Execute Agent is like the ReAct agent but with a focus on planning. It first creates a high-level plan to solve the given task and then executes the plan step by step. This agent is most useful for tasks that require a structured approach and careful planning.

Refer to AI Agent for more information on the AI Agent node itself.

## Node parameters#

Configure the Plan and Execute 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#

Refine the Plan and Execute Agent node's behavior using these options:

### Human Message Template#

Enter a message that n8n will send to the agent during each step execution.

Available LangChain expressions:

• {previous\_steps}: Contains information about the previous steps the agent's already completed.

{previous\_steps}

• {current\_step}: Contains information about the current step.

{current\_step}

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

{agent\_scratchpad}

## 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.