Postgres Chat Memory node documentation

# Postgres Chat Memory node#

Use the Postgres Chat Memory node to use Postgres as a memory server for storing chat history.

On this page, you'll find a list of operations the Postgres Chat Memory node supports, and links to more resources.

Credentials

You can find authentication information for this node here.

Parameter resolution in sub-nodes

Sub-nodes behave differently to other nodes when processing multiple items using an expression.

Most nodes, including root nodes, take any number of items as input, process these items, and output the results. You can use expressions to refer to input items, and the node resolves the expression for each item in turn. For example, given an input of five name values, the expression {{ $json.name }} resolves to each name in turn.

name

{{ $json.name }}

In sub-nodes, the expression always resolves to the first item. For example, given an input of five name values, the expression {{ $json.name }} always resolves to the first name.

name

{{ $json.name }}

## Node parameters#

• Session Key: Enter the key to use to store the memory in the workflow data.

• Table Name: Enter the name of the table to store the chat history in. The system will create the table if doesn't exist.

• Context Window Length: Enter the number of previous interactions to consider for context.

## Related resources#

Refer to LangChain's Postgres Chat Message History documentation for more information about the service.

View n8n's Advanced AI documentation.

## Single memory instance#

If you add more than one Postgres Chat Memory node to your workflow, all nodes access the same memory instance by default. Be careful when doing destructive actions that override existing memory contents, such as the override all messages operation in the Chat Memory Manager node. If you want more than one memory instance in your workflow, set different session IDs in different memory nodes.

## 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.