LangChain concepts in n8n

# LangChain concepts in n8n#

This page explains how LangChain concepts and features map to n8n nodes.

This page includes lists of the LangChain-focused nodes in n8n. You can use any n8n node in a workflow where you interact with LangChain, to link LangChain to other services. The LangChain features uses n8n's Cluster nodes.

n8n implements LangChain JS

This feature is n8n's implementation of LangChain's JavaScript framework.

## Trigger nodes#

Chat Trigger

## Cluster nodes#

Cluster nodes are node groups that work together to provide functionality in an n8n workflow. Instead of using a single node, you use a root node and one or more sub-nodes that extend the functionality of the node.

### Root nodes#

Each cluster starts with one root node.

A chain is a series of LLMs, and related tools, linked together to support functionality that can't be provided by a single LLM alone.

Available nodes:

• Basic LLM Chain

• Retrieval Q&A Chain

• Summarization Chain

• Sentiment Analysis

• Text Classifier

Learn more about chaining in LangChain.

An agent has access to a suite of tools, and determines which ones to use depending on the user input. Agents can use multiple tools, and use the output of one tool as the input to the next. Source

Available nodes:

• Agent

Learn more about Agents in LangChain.

Vector stores store embedded data, and perform vector searches on it.

• Simple Vector Store

• PGVector Vector Store

• Pinecone Vector Store

• Qdrant Vector Store

• Supabase Vector Store

• Zep Vector Store

Learn more about Vector stores in LangChain.

Utility nodes.

LangChain Code: import LangChain. This means if there is functionality you need that n8n hasn't created a node for, you can still use it.

### Sub-nodes#

Each root node can have one or more sub-nodes attached to it.

Document loaders add data to your chain as documents. The data source can be a file or web service.

Available nodes:

• Default Document Loader

• GitHub Document Loader

Learn more about Document loaders in LangChain.

LLMs (large language models) are programs that analyze datasets. They're the key element of working with AI.

Available nodes:

• Anthropic Chat Model

• AWS Bedrock Chat Model

• Cohere Model

• Hugging Face Inference Model

• Mistral Cloud Chat Model

• Ollama Chat Model

• Ollama Model

• OpenAI Chat Model

Learn more about Language models in LangChain.

Memory retains information about previous queries in a series of queries. For example, when a user interacts with a chat model, it's useful if your application can remember and call on the full conversation, not just the most recent query entered by the user.

Available nodes:

• Motorhead

• Redis Chat Memory

• Postgres Chat Memory

• Simple Memory

• Xata

• Zep

Learn more about Memory in LangChain.

Output parsers take the text generated by an LLM and format it to match the structure you require.

Available nodes:

• Auto-fixing Output Parser

• Item List Output Parser

• Structured Output Parser

Learn more about Output parsers in LangChain.

• Contextual Compression Retriever

• MultiQuery Retriever

• Vector Store Retriever

• Workflow Retriever

Text splitters break down data (documents), making it easier for the LLM to process the information and return accurate results.

Available nodes:

• Character Text Splitter

• Recursive Character Text Splitter

• Token Splitter

n8n's text splitter nodes implements parts of LangChain's text\_splitter API.

Utility tools.

• Calculator

• Code Tool

• SerpAPI

• Think Tool

• Vector Store Tool

• Wikipedia

• Wolfram|Alpha

• Workflow Tool

Embeddings capture the "relatedness" of text, images, video, or other types of information. (source)

Available nodes:

• Embeddings AWS Bedrock

• Embeddings Cohere

• Embeddings Google PaLM

• Embeddings Hugging Face Inference

• Embeddings Mistral Cloud

• Embeddings Ollama

• Embeddings OpenAI

Learn more about Text embeddings in LangChain.

• Chat Memory Manager