

## No Code with n8n





Facebook somkiat.cc

Page Messages Notifications 3 Insights Publishing Tools Settings Help

somkiat.cc  
@somkiat.cc

Home Posts Videos Photos

Liked Following Share ... + Add a Button



# Topics

Introduction to Automation and No Code Tools  
RAG workflow  
Working with n8n  
Data ingestion and chunking  
Data embedding  
Store in Vector database  
Workshop with real-world use cases



# Automation

## The key to be data-driven



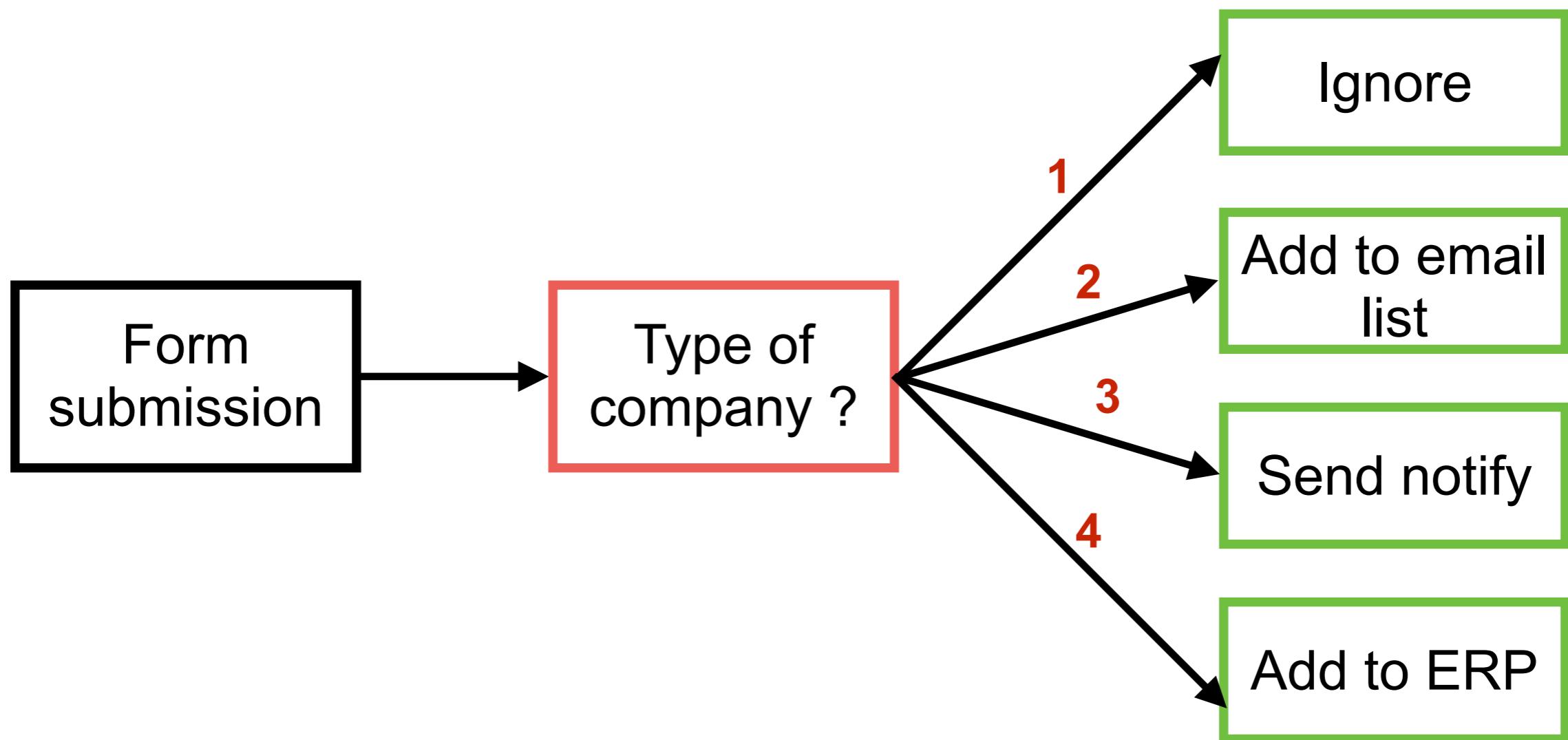
# Manual tasks ?

Wasted time  
Human error  
Repeated low value tasks  
High human resource requirements  
Low employee happiness and retention



# Automation ?

Predictable set of predetermined actions  
that transfers data from one point to another



# Automation ?

Predictability

Better data availability

Increased employee efficiency

Focus efforts on higher-value tasks

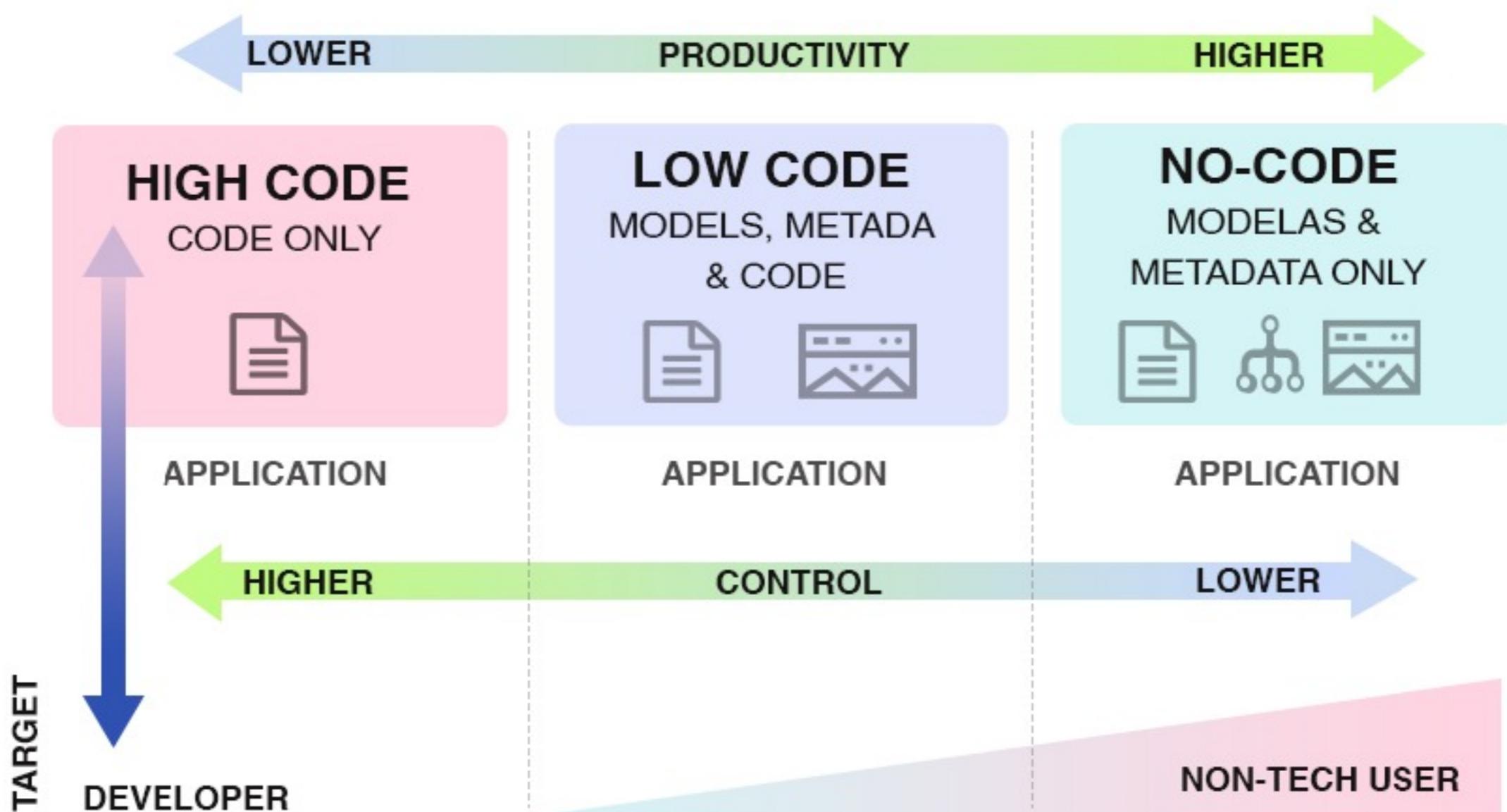
Much higher ROI (Return in Investment)



# NoCode Tools ?



# High code vs Low Code vs No Code





n8n

-zapier

 **make**  
formerly Integromat

 **Langflow**





# NodematioN

<https://n8n.io/>



# NodematioN



## Flexible AI workflow automation for technical teams

Build with the precision of code or the speed of drag-n-drop. Host with on-prem control or in-the-cloud convenience. n8n gives you more freedom to implement multi-step AI agents and integrate apps than any other tool.

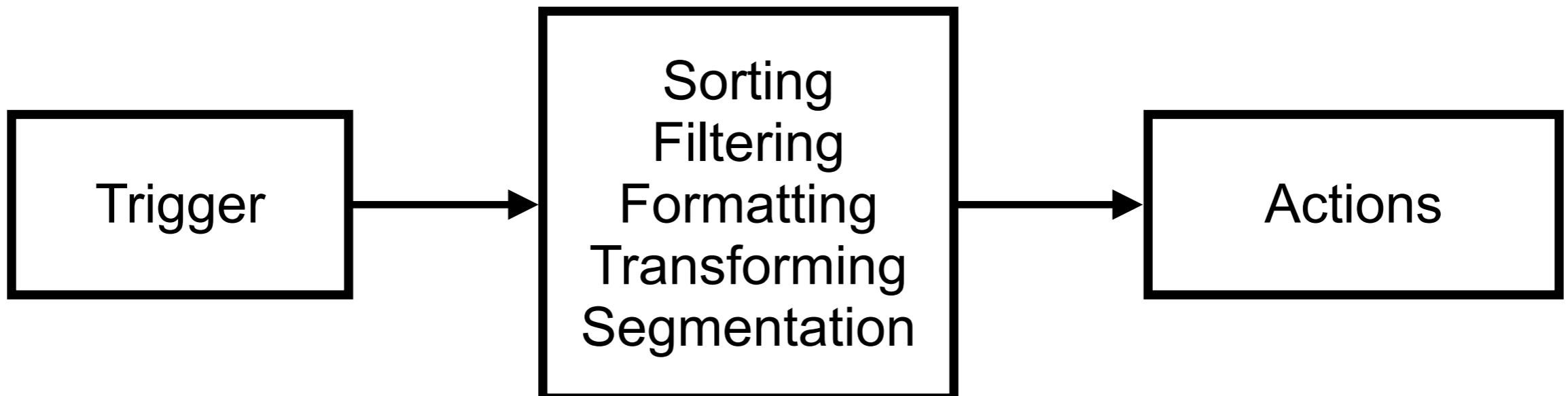
[Get started for free](#)[Talk to sales](#)

<https://n8n.io/>



# n8n ?

## Workflow automation

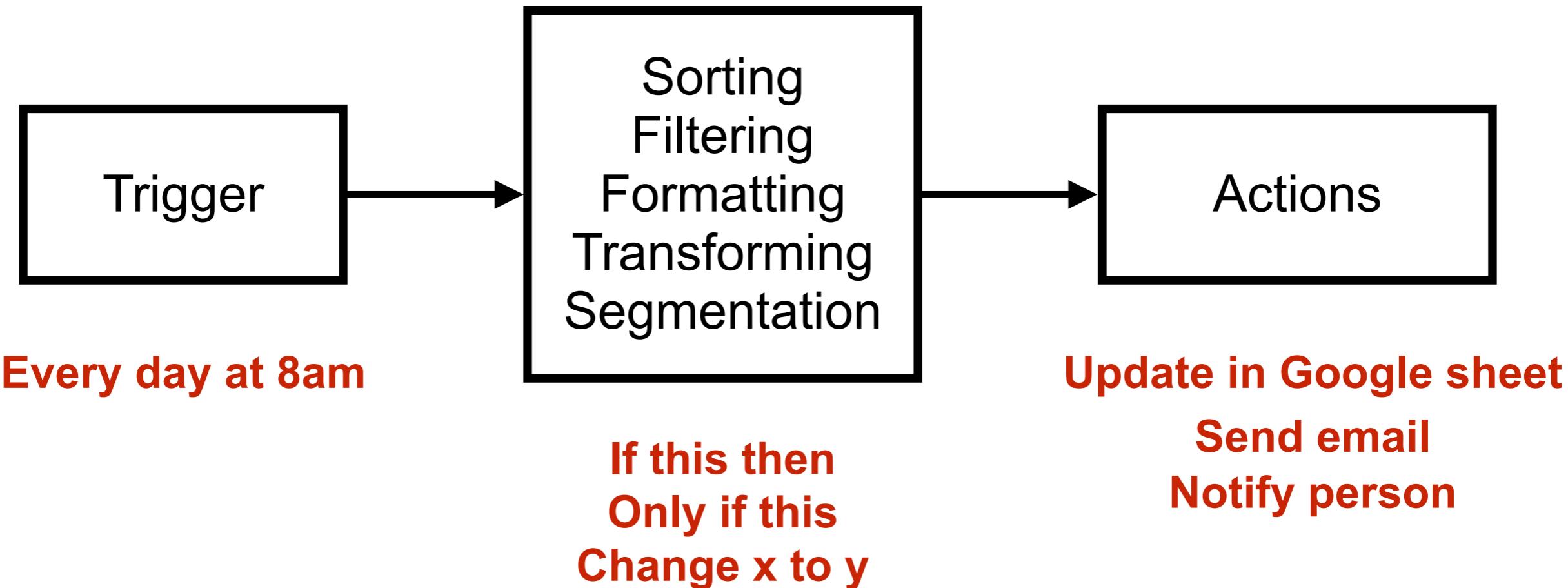


<https://n8n.io/>



# n8n ?

## Workflow automation



# Trigger ?

Start point of automation workflow

Manual

Scheduled

Applications

**Every minute**

**Every day at 8 am**

**Every end of month**

**Webhook or callback**

**Property update**

**Form submission**



# Schedule expression

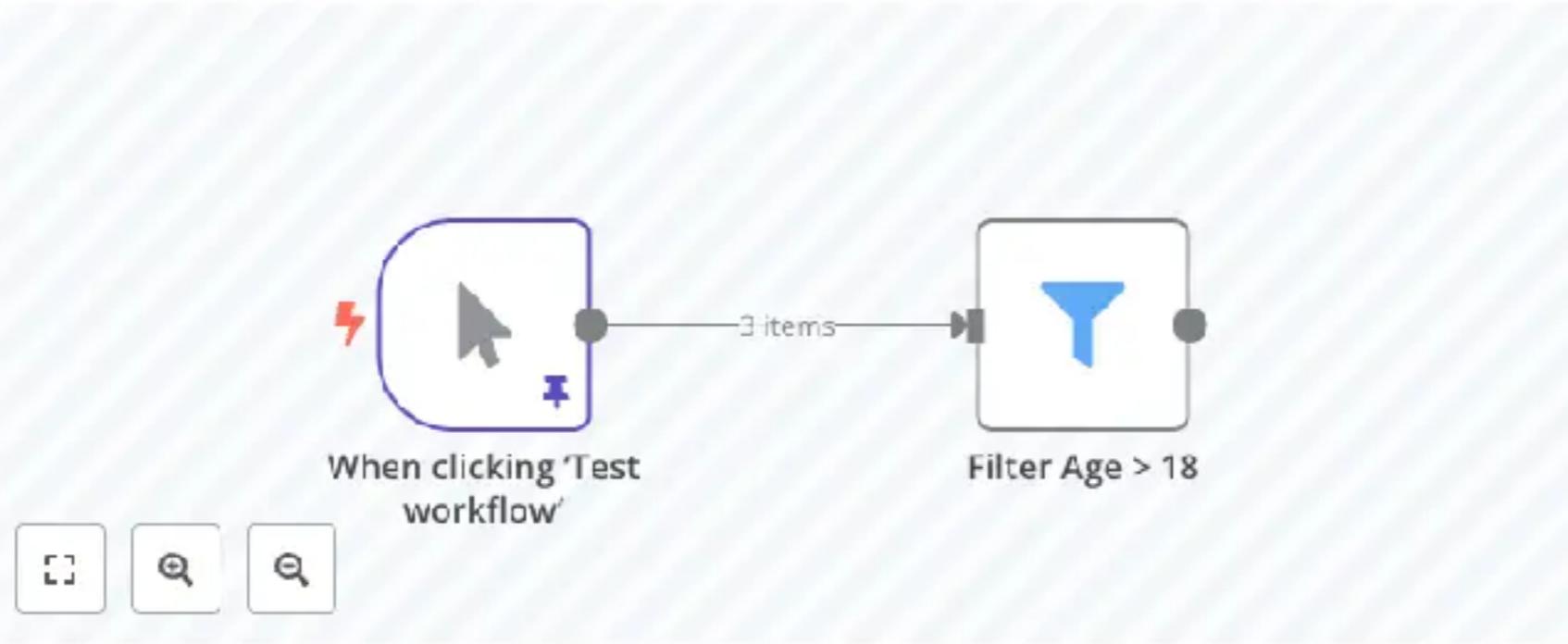
# crontab guru

<https://crontab.guru/>



# Filtering ?

Used to allow or block certain types of data from following a path based on conditions



# Actions ?

Allow to interact with applications

Google Sheet

LINE  
Messaging

Email

Database

Discord

ERP



# n8n Cheat Sheet

[https://x.com/n8n\\_io/status/1950828880127746464/photo/1](https://x.com/n8n_io/status/1950828880127746464/photo/1)



# Workflow templated

4128 Workflow Automation Templates

Search apps, roles, usecases...

AI Sales IT Ops Marketing Document Ops Other Support

Newcomer essentials: learn by doing

Build Your First AI Agent

Lucas Peyrin 0

Your First AI Agent

Chat Model Memory Tool

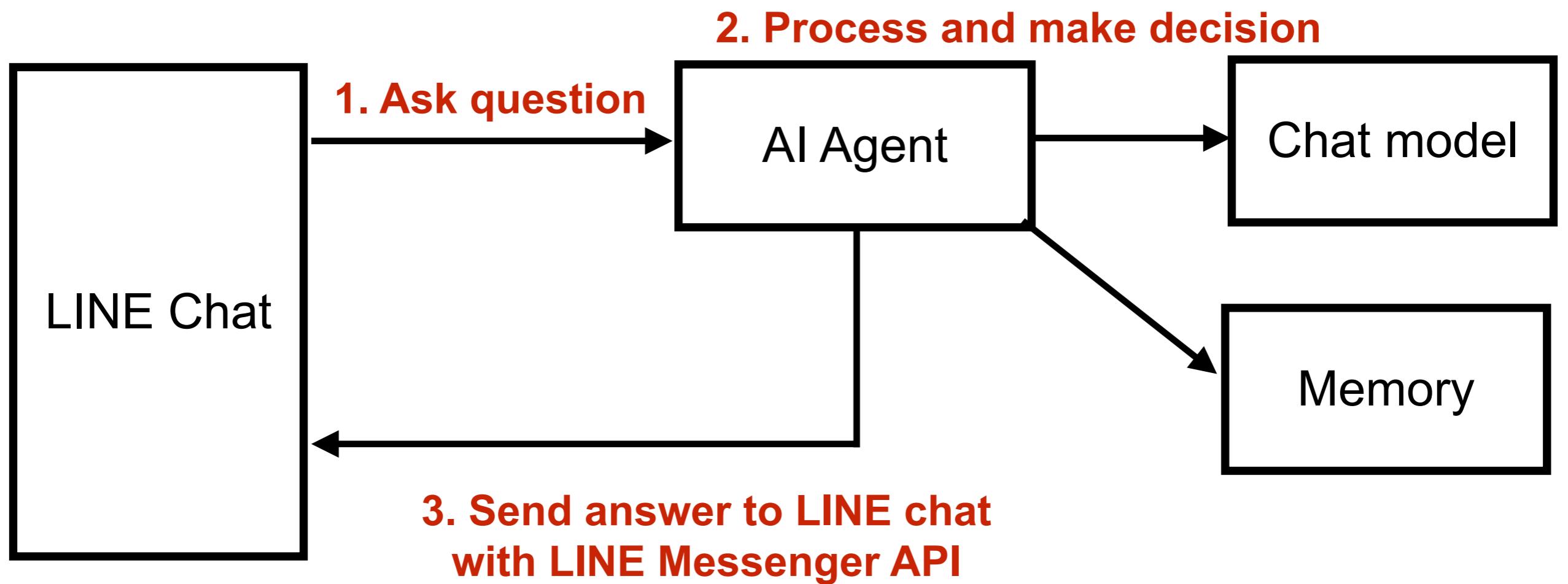
<https://n8n.io/workflows/>



# **Start with your workflow !!**



# ChatBot in LINE



# Step to build AI chatbot

Define your chatbot's purpose

Identify your target audience

Identify your use cases

Design conversation flow

Choose LLM provider

Connect to external data sources

Choose development tools

Test, refine and deploy

<https://blog.n8n.io/how-to-make-ai-chatbot/>



# Let's start



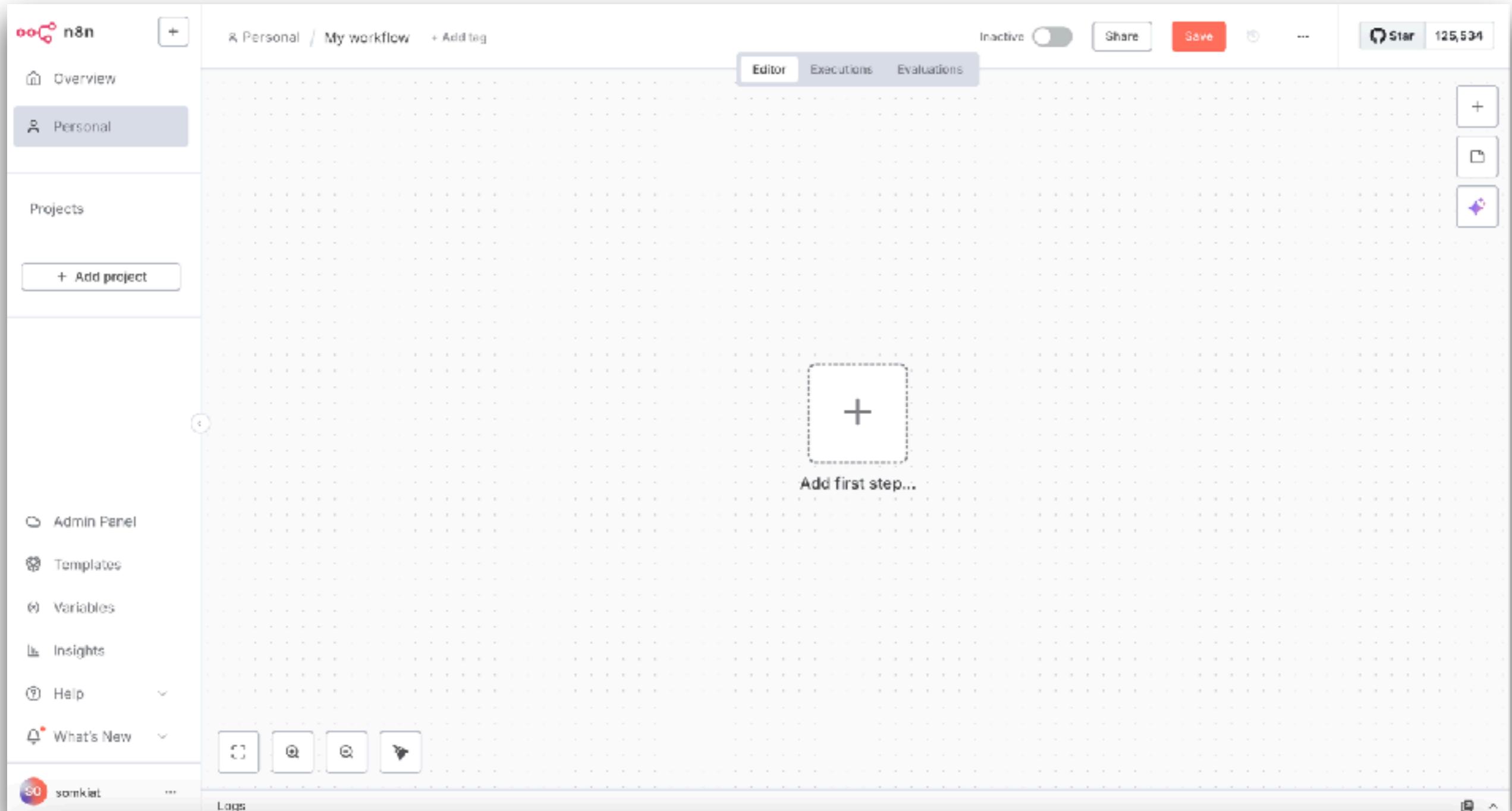
# Installation

On-Cloud

On-Premise



# n8n On-Cloud

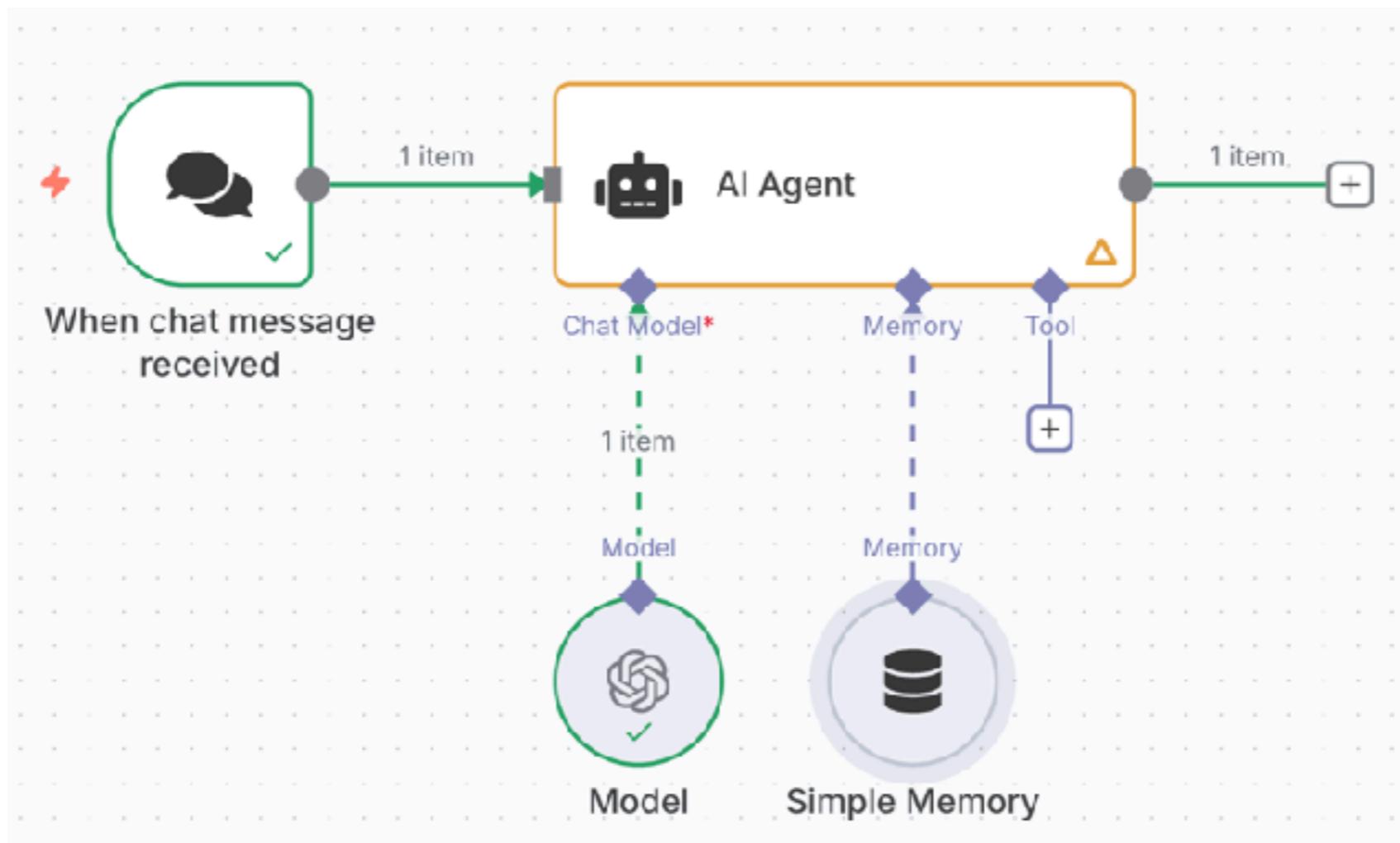


<https://n8n.io/>



# Chat with AI Agent

ສື່ເລື້ອງ ?

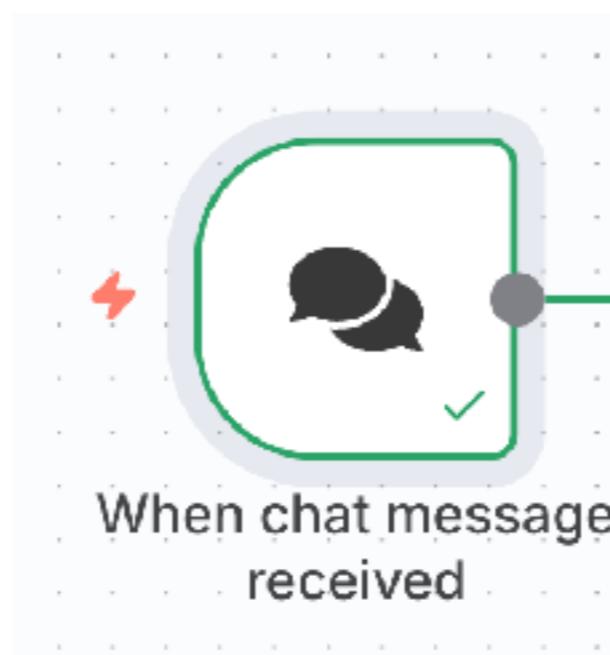


<https://github.com/up1/workshop-ai-with-technical-team/wiki/Prompt-Engineering>



# 1. Chat Trigger Node

This node listens for incoming message and start chat

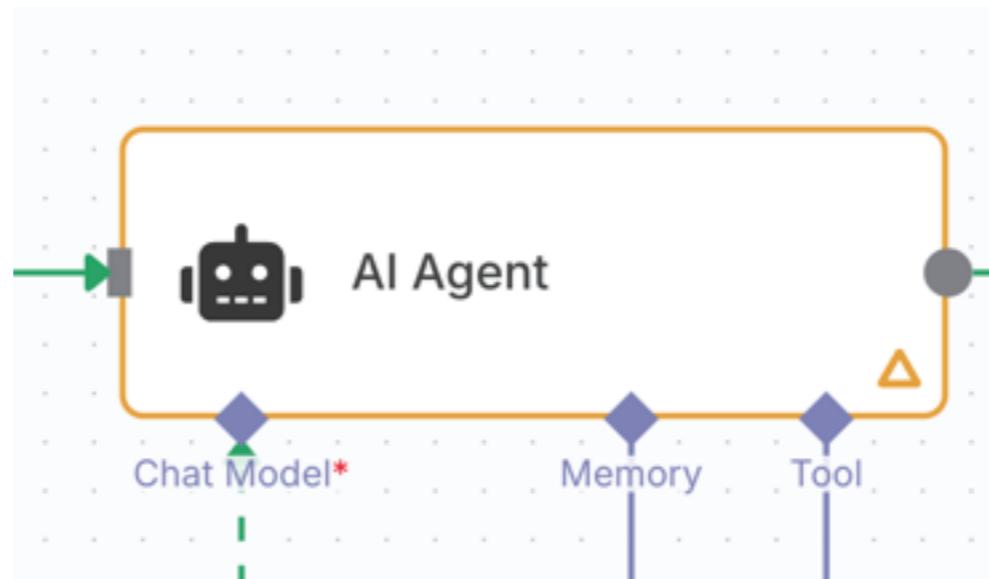


<https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-langchain.chattrigger/>



## 2. Connect to AI Agent Node

This node serves as Decision-maker  
Provide system prompt

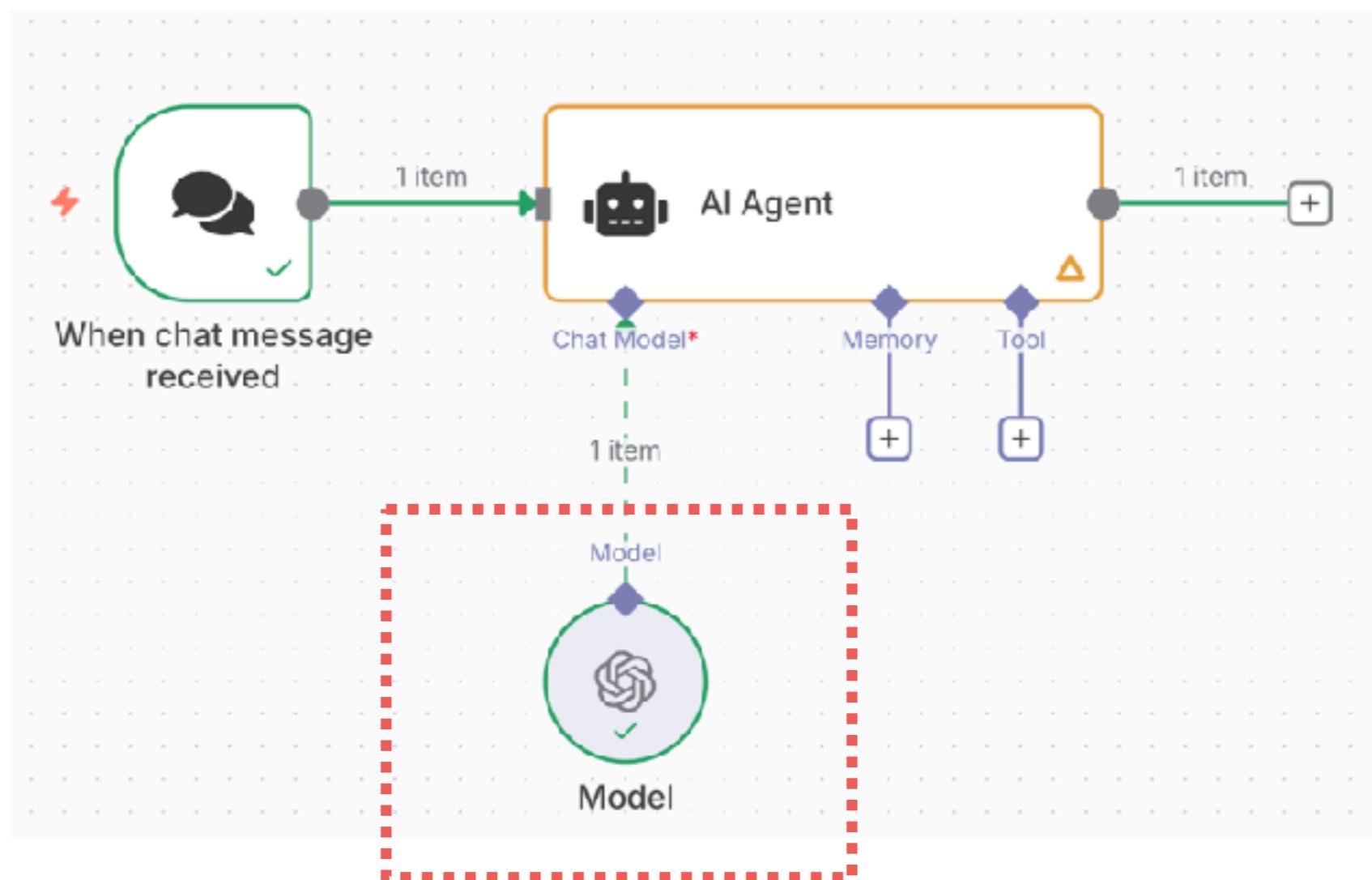


<https://docs.n8n.io/integrations/builtin/cluster-nodes/root-nodes/n8n-nodes-langchain.agent/>



# 3. Integrate with Chat Model

Choose LLM/AI provider

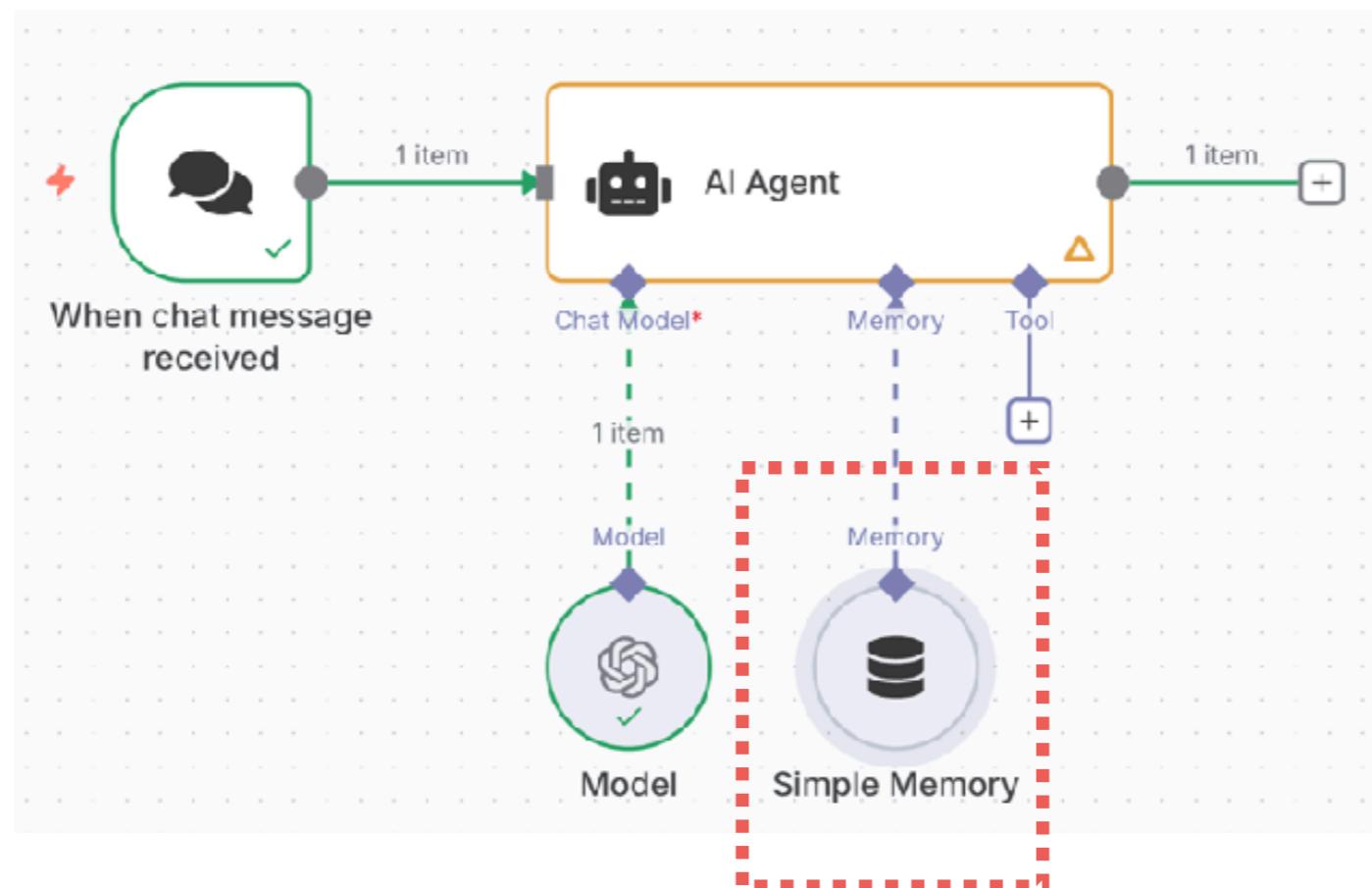


<https://docs.n8n.io/integrations/builtin/cluster-nodes/root-nodes/n8n-nodes-langchain.agent/>



# 4. Add Memory node for context

Use simple memory node  
Maintain conversation context, store last messages



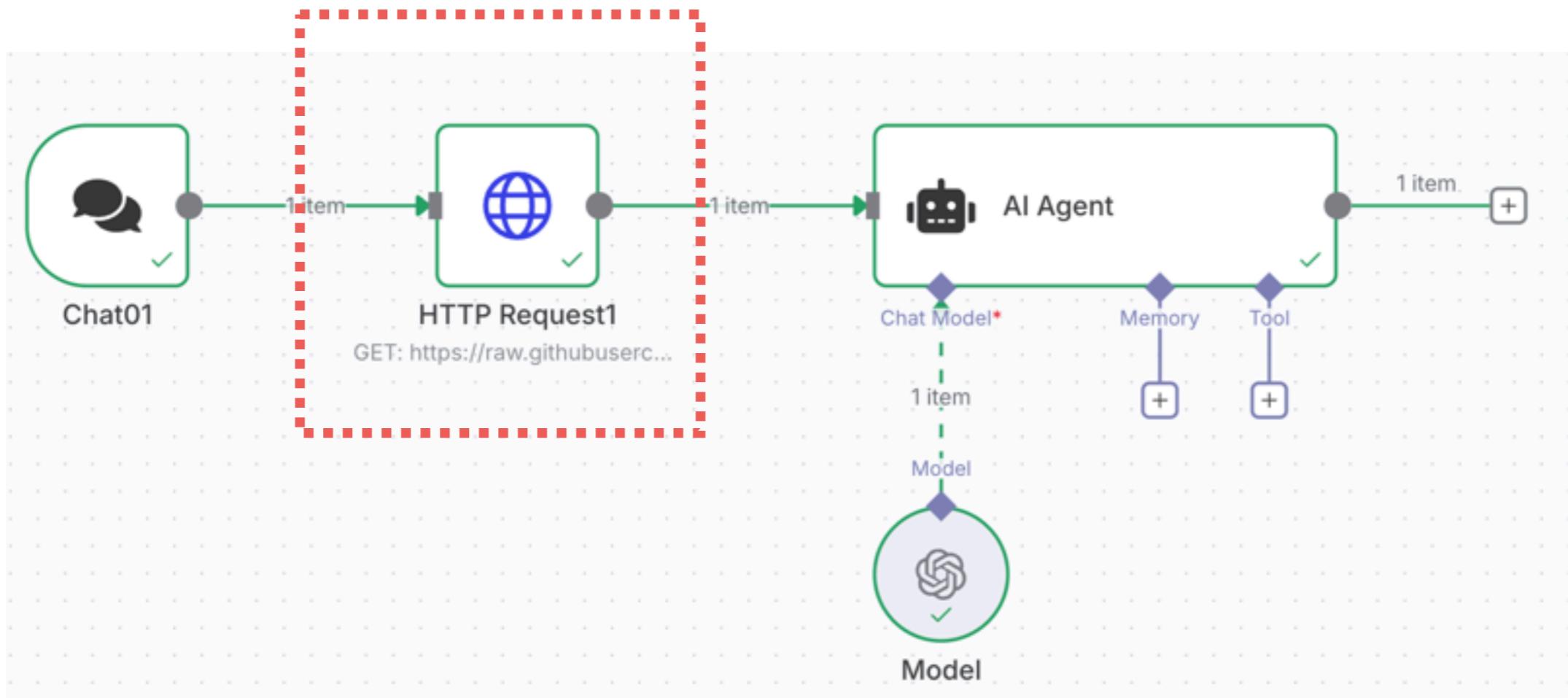
<https://docs.n8n.io/integrations/builtin/cluster-nodes/sub-nodes/n8n-nodes-langchain.memorybufferwindow/>



# Try by yourself



# Read data from web ?



<https://raw.githubusercontent.com/up1/workshop-n8n/refs/heads/main/workshop/data/joke.txt>



# Edit AI agent node !!

AI Agent

Parameters    Settings    Docs

1. Data from Chat node

Source for Prompt (User Message)  
Define below

Prompt (User Message)  
fx {{node['Chat01'].json.chatInput}}

Require Specific Output Format

Enable Fallback Model

Options

2. Data from HTTP request node

System Message

Answer the question as truthfully as possible using the provided text, and if the answer is not contained within the text below, say "I don't know"  
fx {{ node['HTTP Request1'].json.data }}

Are you ready to talk with me ?

Result  
Instruction:  
Answer the question as truthfully as possible using the provided text, and if the answer is not contained within the text below, say "I don't know"

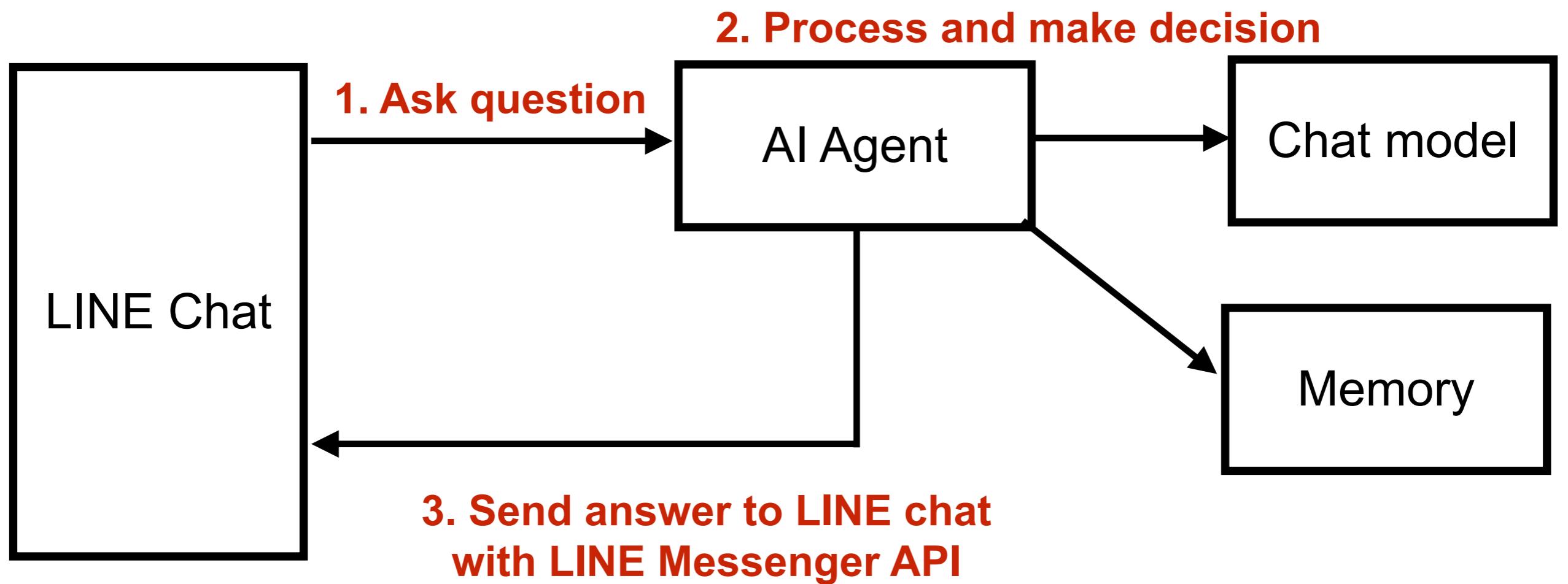
Chat Model    Memory    Tool



# ChatBot in LINE

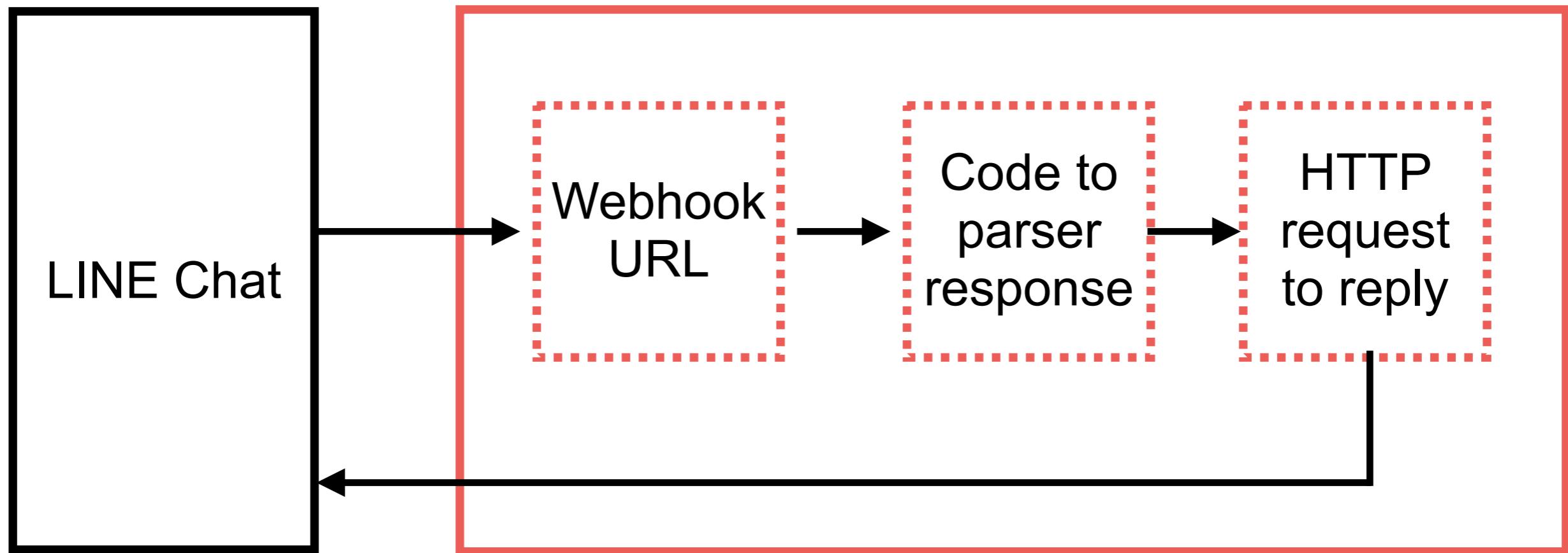


# ChatBot in LINE



# ChatBot in LINE with n8n

## Workflow in n8n



# Working with LINE

## Messaging API

Get engaged in interactive conversations with LINE users.

### What is the Messaging API?

The Messaging API lets you develop two-way communication between your service and LINE users.



[Start now](#)

[Documentation](#)

<https://developers.line.biz/en/services/messaging-api/>



# Create a Messaging API channel

This provider doesn't have any channels yet

To create one, choose a channel type below



Create a LINE Login channel



Create a Messaging API channel



Create a Blockchain Service channel



Create a LINE MINI App channel

Available features might defer based on the account with which you are currently logged in. For details, check the [document](#).

<https://developers.line.biz/en/services/messaging-api/>



# Enable Messaging API

## Messaging API

Messaging API is an advanced feature for developers. It allows accounts to promote more interactive communication by sending messages.

[What is Messaging API?](#)

[LINE Developers API documentation](#)

Status	Enabled		
Channel info	Channel ID	2007859459	<a href="#">Copy</a>
	Channel secret	a695a88c8cc706557182527898f5a068	<a href="#">Copy</a>
Webhook URL	<input type="text" value="https://"/>	<a href="#">Save</a>	

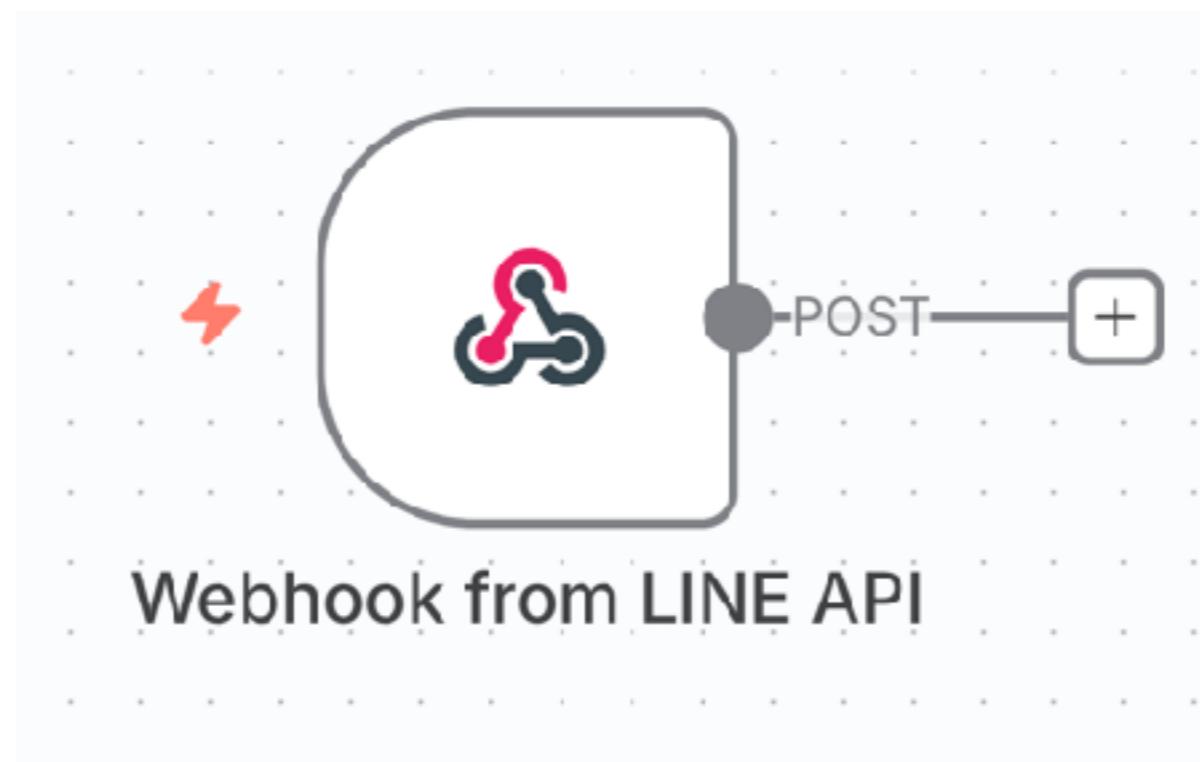
Add web hook URL ?

<https://developers.line.biz/en/services/messaging-api/>



# Create Web hook node

Receive data from apps and services when event occurs  
Trigger node



<https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.webhook/>



# Webhook node

The screenshot shows the configuration interface for a 'Webhook from LIN...' node. At the top, there's a red button labeled 'Listen for test event'. Below it, tabs for 'Parameters' (which is selected), 'Settings', and 'Docs' are visible. A red dashed box highlights the 'Webhook URLs' section. Inside this section, there are two tabs: 'Test URL' (selected) and 'Production URL'. Under 'Test URL', a 'POST' method is selected, and the URL is shown as `https://somkiatspns.app.n8n.cloud/webhook-test/c2ab6221-0464-4eaa-ac9a-3a208f98147a`. Below this, the 'HTTP Method' is set to 'POST', and the 'Path' is set to `c2ab6221-0464-4eaa-ac9a-3a208f98147a`. The 'Authentication' dropdown is set to 'None', and the 'Respond' dropdown is set to 'Immediately'.

**URL of webhook**

<https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.webhook/>



# Test from LINE

The screenshot shows the n8n interface for testing a webhook from the LINE API. On the left, the configuration pane shows the following details:

- Webhook URLs:** Test URL: <https://somklatspns.app.n8n.cloud/webhook-test/c2ab6221-0464-4eaa-ac9a-3a208f98147a>
- HTTP Method:** POST
- Path:** c2ab6221-0464-4eaa-ac9a-3a208f98147a
- Authentication:** None
- Respond:** Immediately
- Options:** No properties

On the right, the test results pane shows the following output:

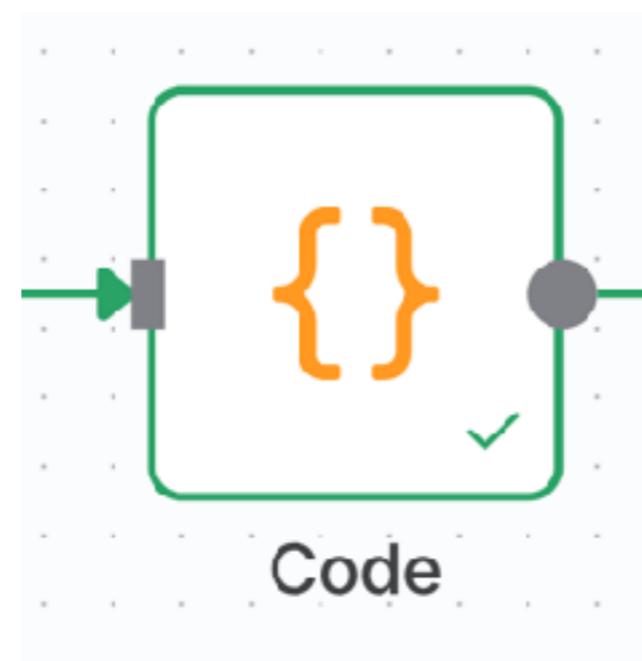
```
OUTPUT ⓘ
1 item
[{"headers": {"host": "somklatspns.app.n8n.cloud", "user-agent": "LineBotNebhook/2.0", "content-length": "63", "accept-encoding": "gzip, br", "cdn-loop": "cloudflare; loops=1; subreqs=1", "cf-connecting-ip": "147.92.149.166", "cf-ew-via": "15", "cf-ipcountry": "JP", "cf-ray": "967ed7ecc332cfb2-NRT", "cf-visitor": "{\"scheme\": \"https\"}", "cf-worker": "n8n.cloud", "content-type": "application/json; charset=utf-8", "x-forwarded-for": "147.92.149.166, 172.70.223.173", "x-forwarded-host": "somklatspns.app.n8n.cloud", "x-forwarded-port": "443", "x-forwarded-proto": "https", "x-forwarded-server": "traefik-prod-users-gw-47-58bccd9445-cbh7b", "x-is-trusted": "yes", "x-line-signature": "u/ZXham6m6iAMFXuLB378xI8zSNCXUWj8UdD4ZXXluY=", "x-real-ip": "147.92.149.166"}, "params": {}, "query": {}}
```

<https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.webhook/>



# Code node

Write custom JavaScript or Python code  
Parse JSON data



<https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.code>



# Code node

The screenshot shows the n8n interface with a 'Code' node selected. The 'INPUT' tab on the left displays a complex JSON object representing a webhook from the LINE API. The 'OUTPUT' tab on the right shows a simplified JSON object with 'text' and 'replyToken' fields. The central 'Code' tab contains a JavaScript code block:

```
let requests
let message = {}
for (const item of $input.all()) {
    requests = item.json.body
}
if ('events' in requests) {
    if ('message' in requests['events'][0]) {
        if (requests['events'][0]['message']['type'] == 'text') {
            'text' in requests['events'][0]['message']) {
                message = {
                    'text': requests['events'][0]['message']['text'],
                    'replyToken': requests['events'][0]['replyToken']
                }
            }
        }
    }
}
return message
```

A note at the bottom of the code editor says: "Type \$ for a list of **special vars/methods**. Debug by using console.log() statements and viewing their output in the browser console."

<https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.code>

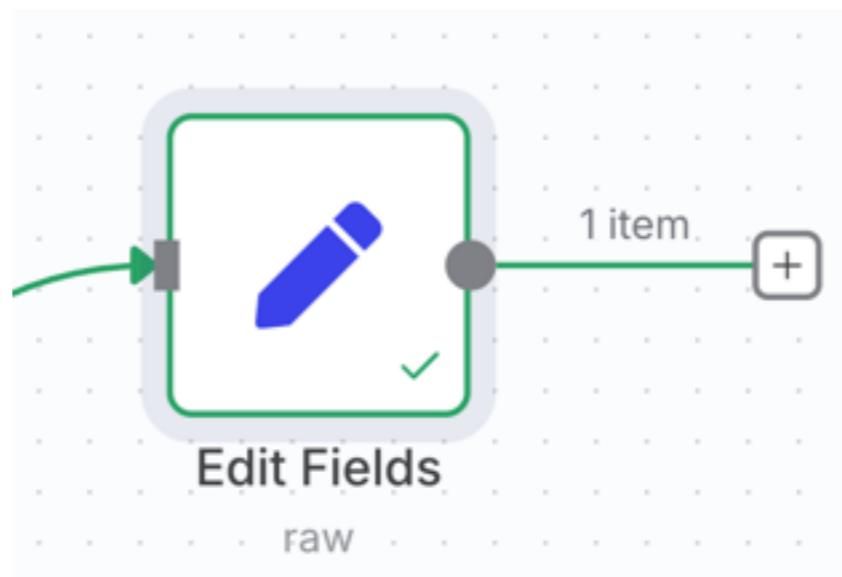


# No Code !!



# Edit field

Transform input data to new format  
Parse JSON data



<https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.set/>



# Edit field

 Edit Fields Execute step

[Parameters](#) [Settings](#) [Docs ↗](#)

Mode  
JSON

JSON

```
{  
    "chatInput": "{{ $json.body.events[0].message.text }}",  
    "replyToken": "{{ $json.body.events[0].replyToken }}"  
}
```

{ "chatInput": "Mango", "replyToken": "479ba152eb364623a2e755feb07e5c74" }

Include Other Input Fields

Options

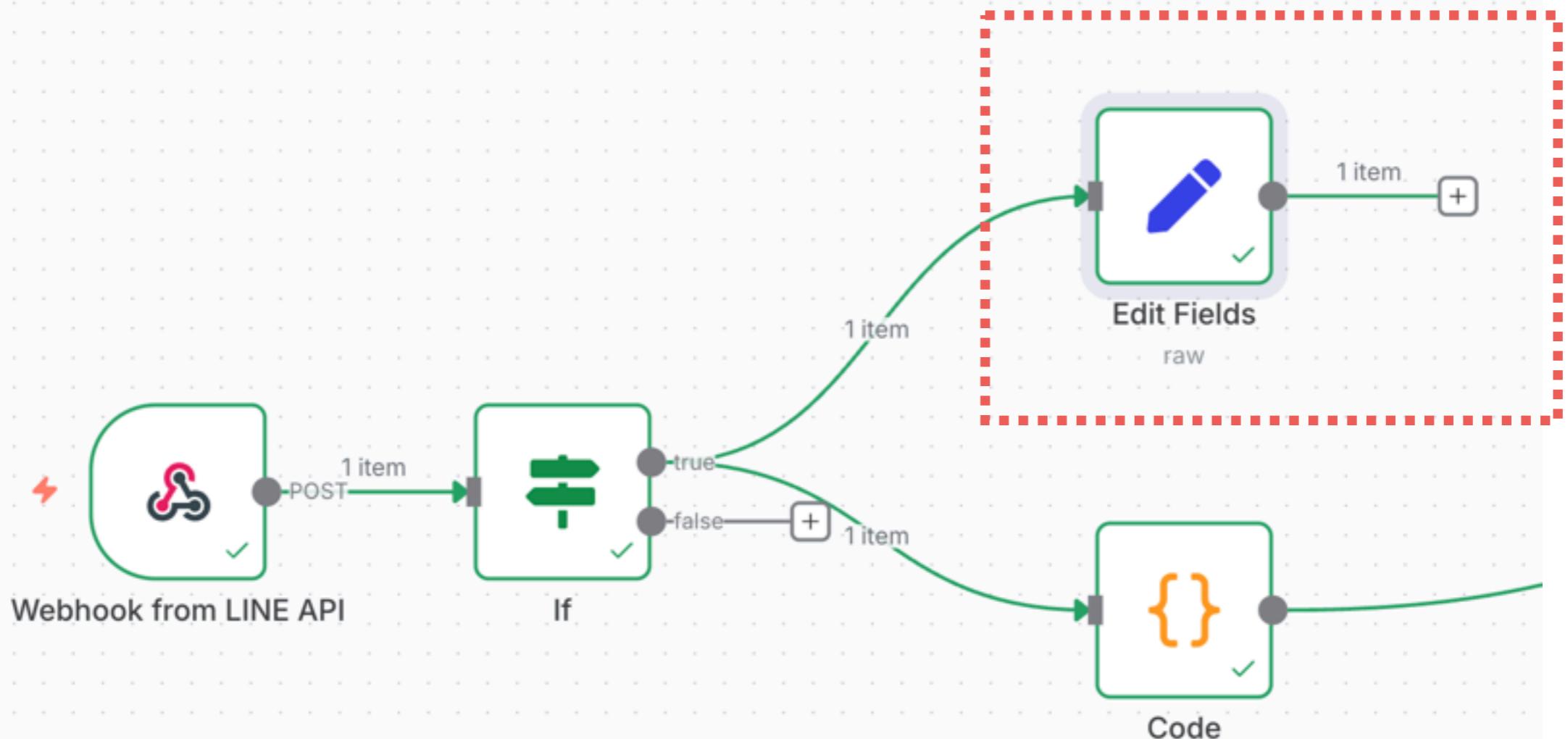
No properties

Add option

<https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.set/>

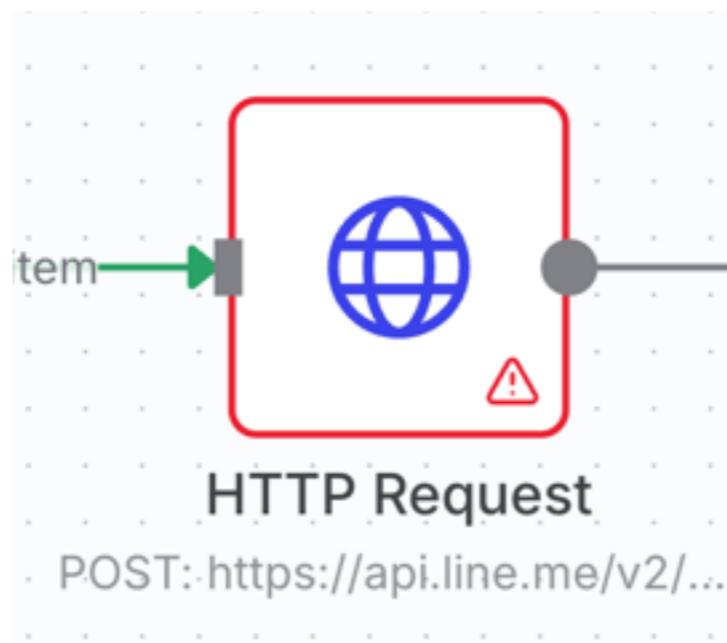


# Add Edit field to workflow



# HTTP Request node

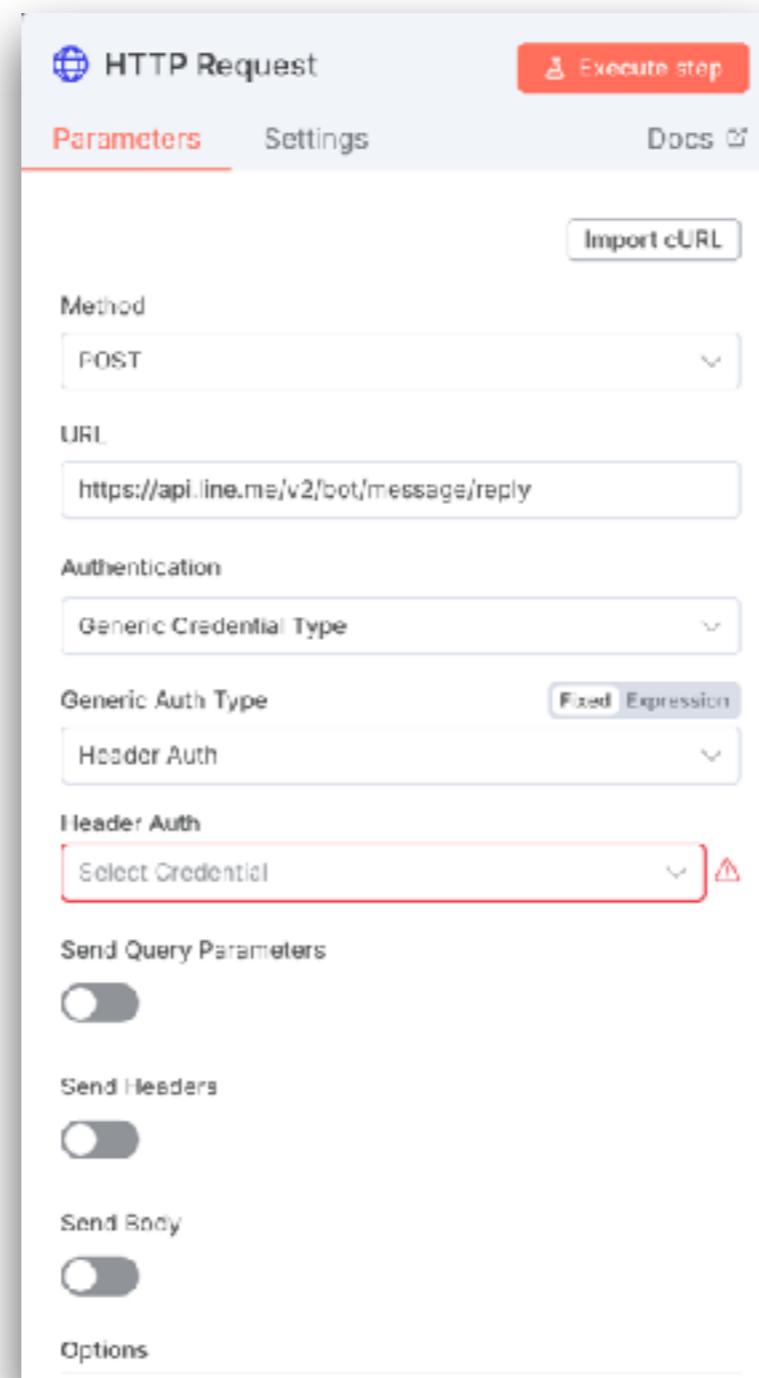
Make HTTP requests to target apps/services  
Call REST API



<https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.httprequest>



# Send to LINE API



<https://api.line.me/v2/bot/message/reply>



# Header Auth

Header Auth account  
Header Auth

Save X

Connection Need help filling out these fields? [Open docs](#)

Sharing [Ask Assistant](#) for setup instructions

Details

Name Fixed Expression

Authorization

Value

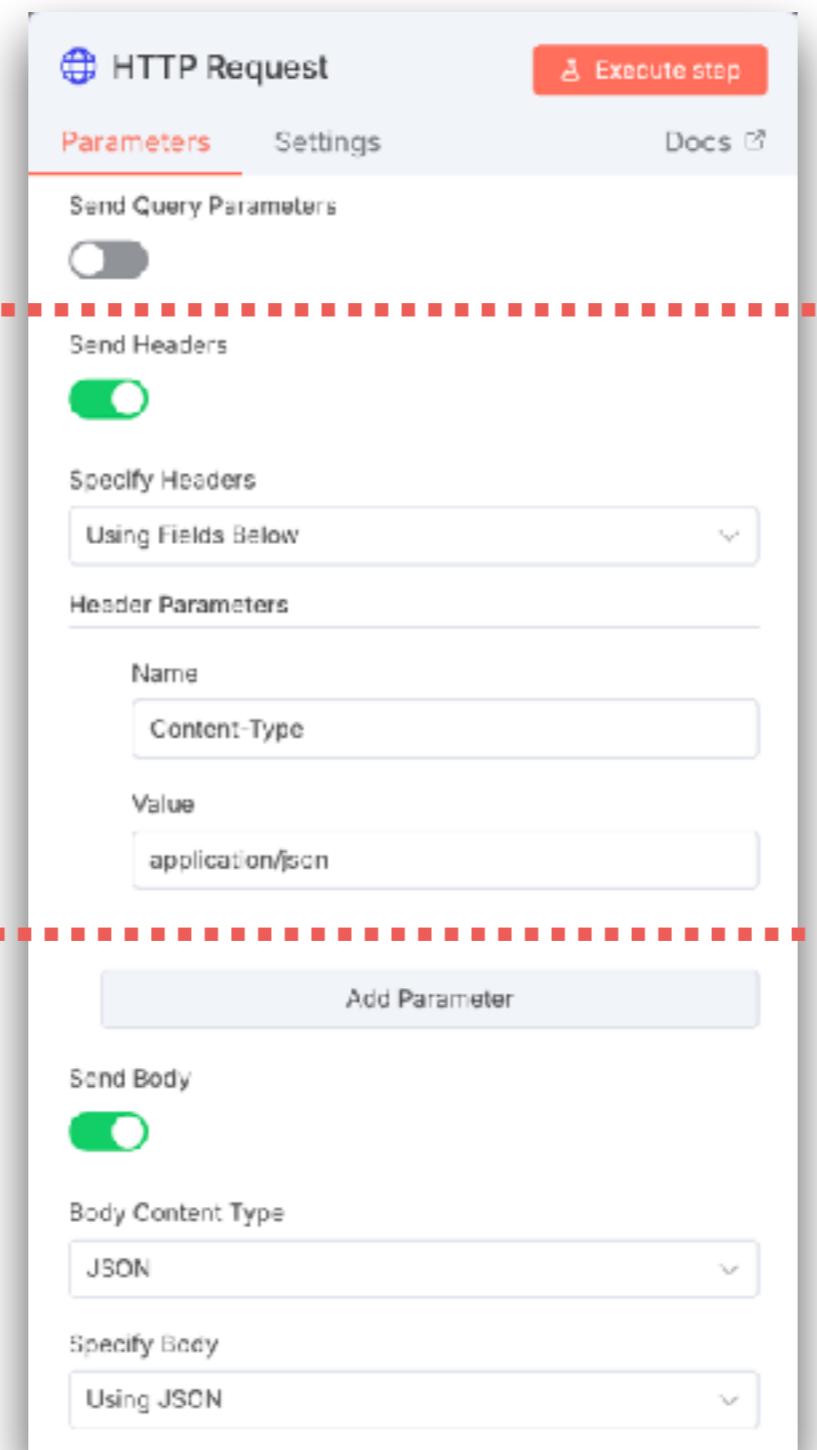
..... **Bearer <line token>** .....

To send multiple headers, use a "Custom Auth" credential instead

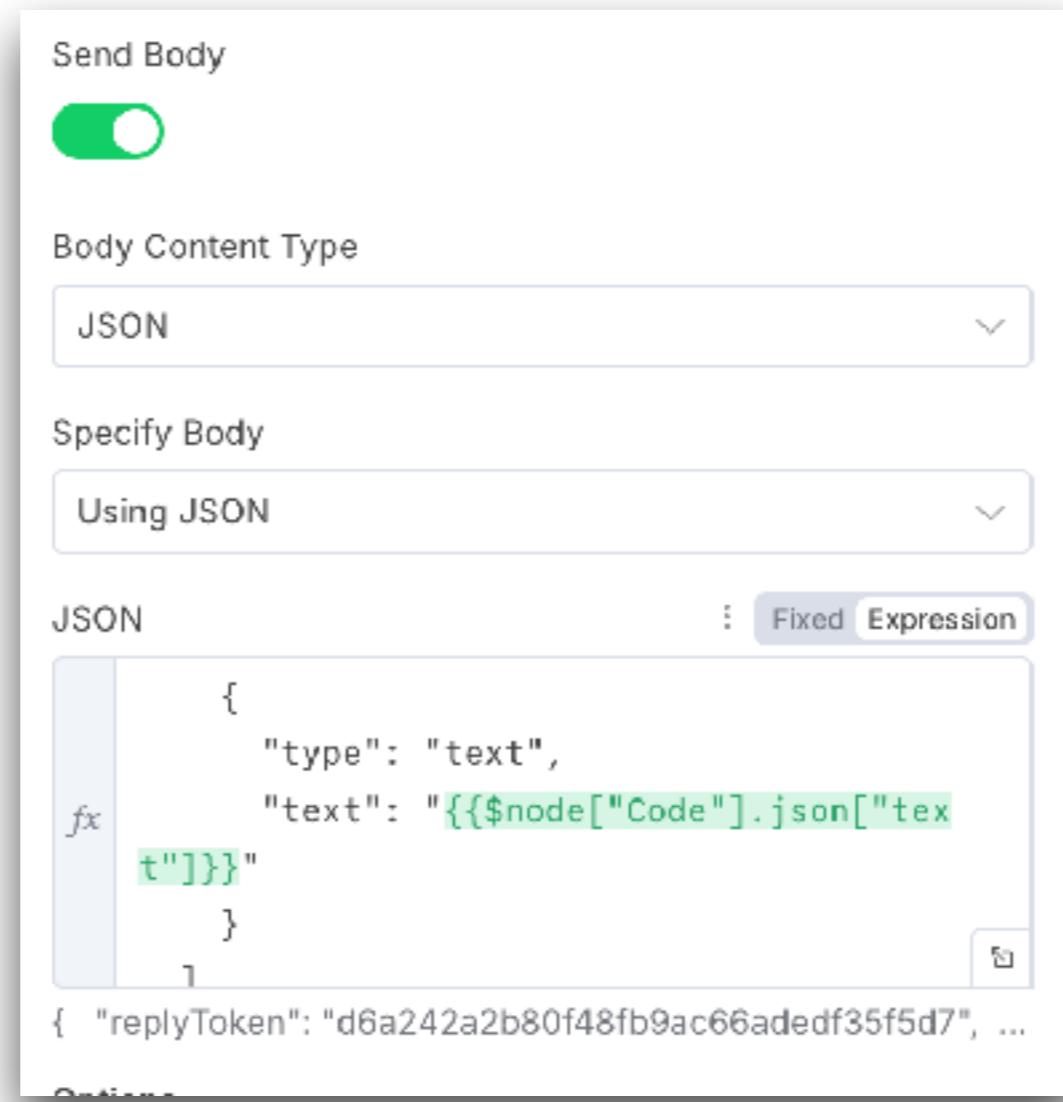
ⓘ Enterprise plan users can pull in credentials from external vaults. [More Info](#)



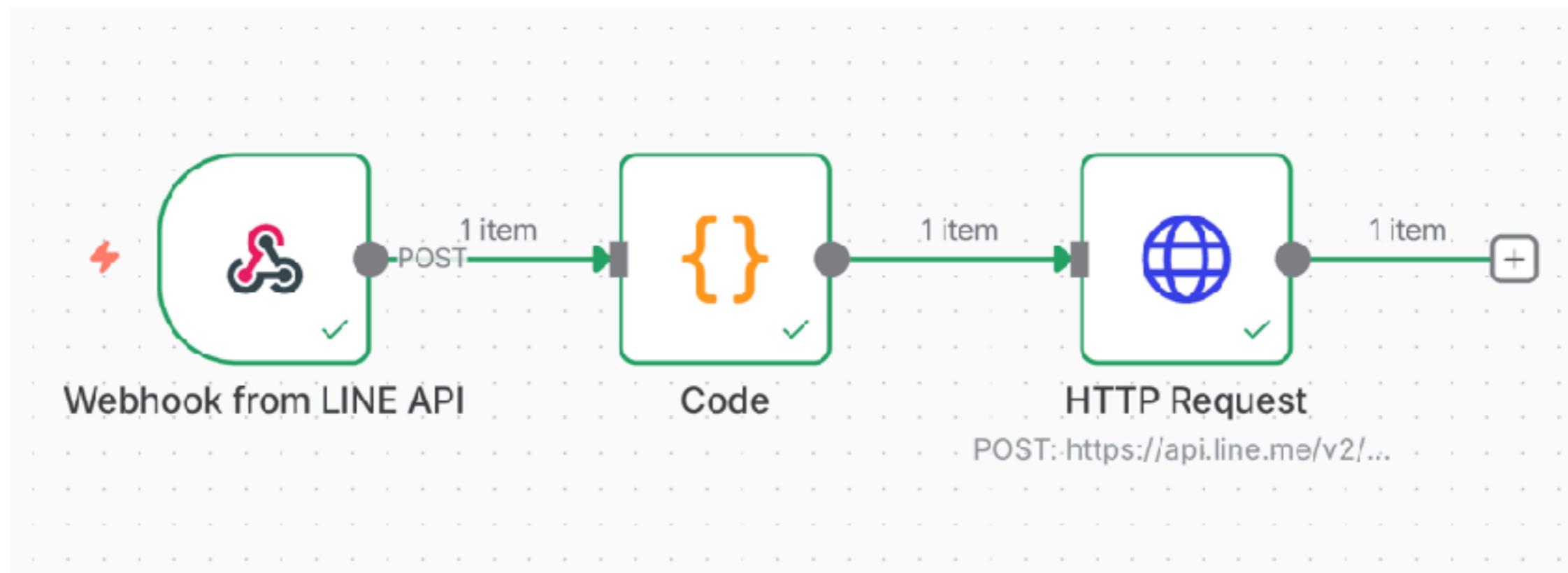
# Customer header



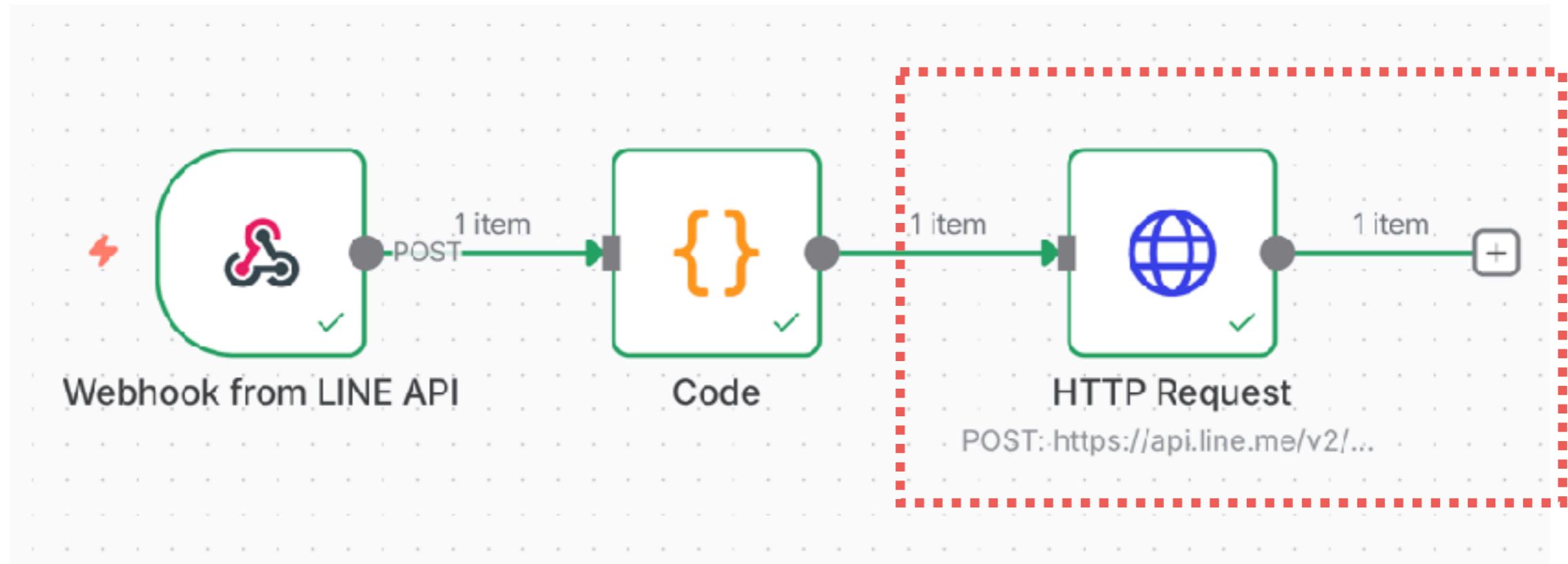
# Customer body



# Receive and reply message



# Reply message with LINE API



# Reply message with LINE API

The screenshot shows the LINE Developers documentation website. The top navigation bar includes links for LINE Developers, About, News, Products, Documentation (underlined), FAQ, Glossary, More, a search icon, and a language switcher. On the left, there's a sidebar with sections for API reference, Development guidelines, Quickstart, and Guides. Under Guides, the 'Send messages' section is highlighted with a green background. Below the sidebar, the main content area has a heading 'Send messages'. To the left of the main content, there's a sidebar with links for Character counting in a text, Get user IDs, and Stickers. The main content area contains text about the Messaging API and a list of messaging types.

With the Messaging API, you can make your bot send messages to users. At any time, you can initiate to send a message to a user and reply to a user's message. You can also use various message types.

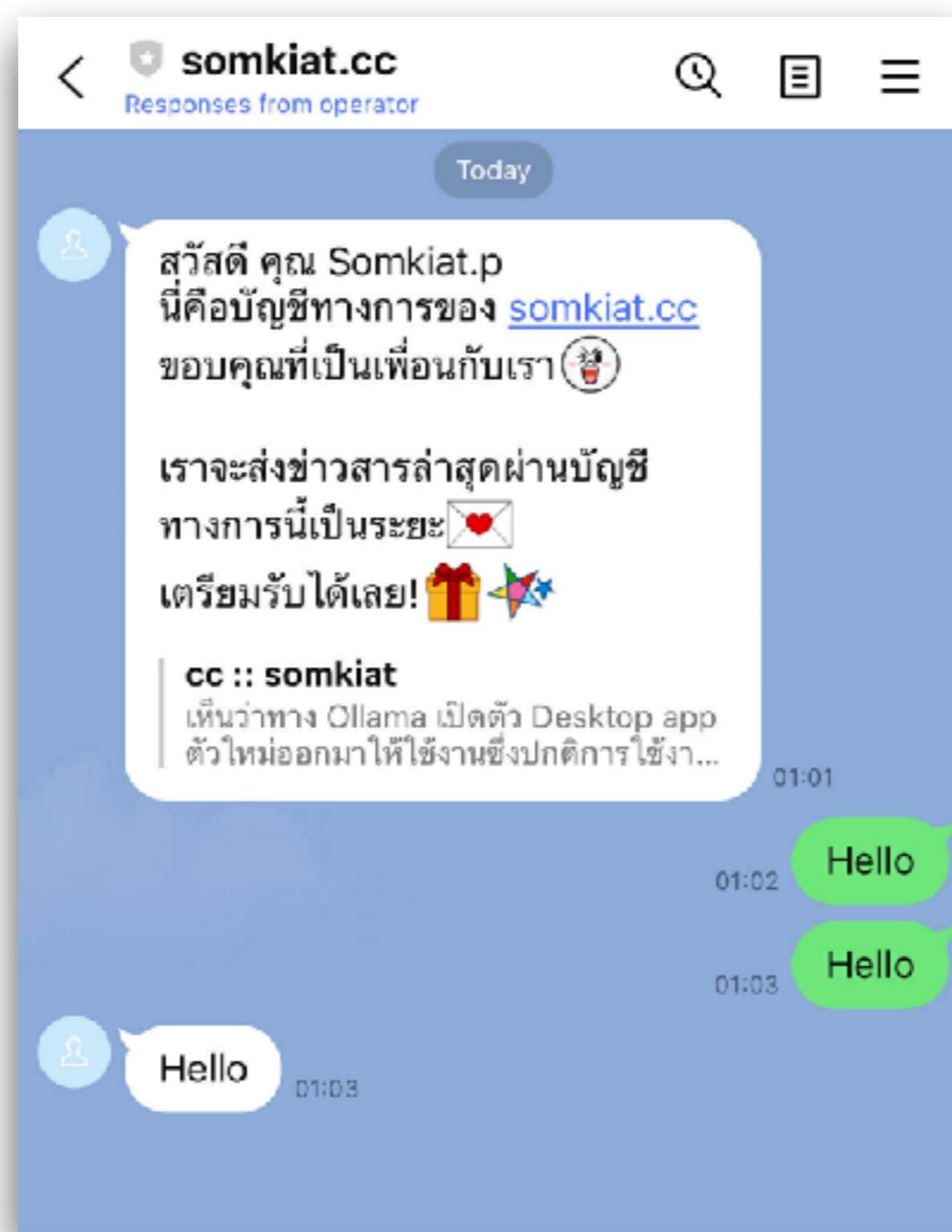
Messaging types

- Reply message
- Push message: One-to-one
- Multicast message: One-to-many (Targets a list of user IDs)
- Narrowcast message: One-to-many (Targets a segmented list)
- Broadcast message: One-to-many (Targets all friends)

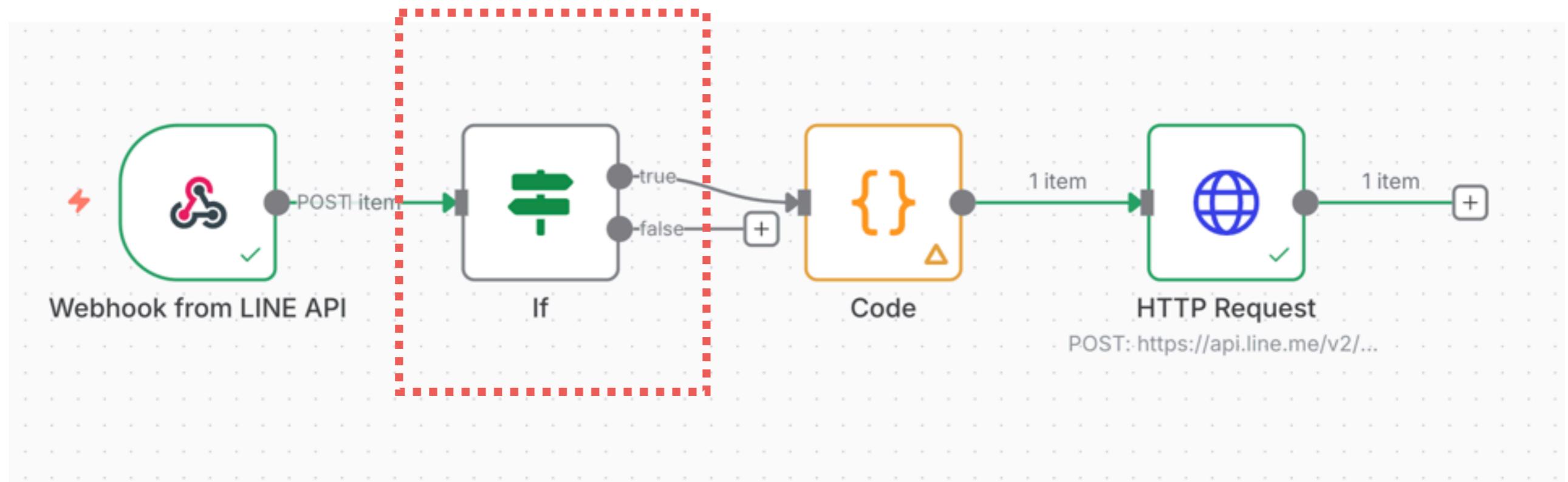
<https://developers.line.biz/en/docs/messaging-api/sending-messages/>



# Example

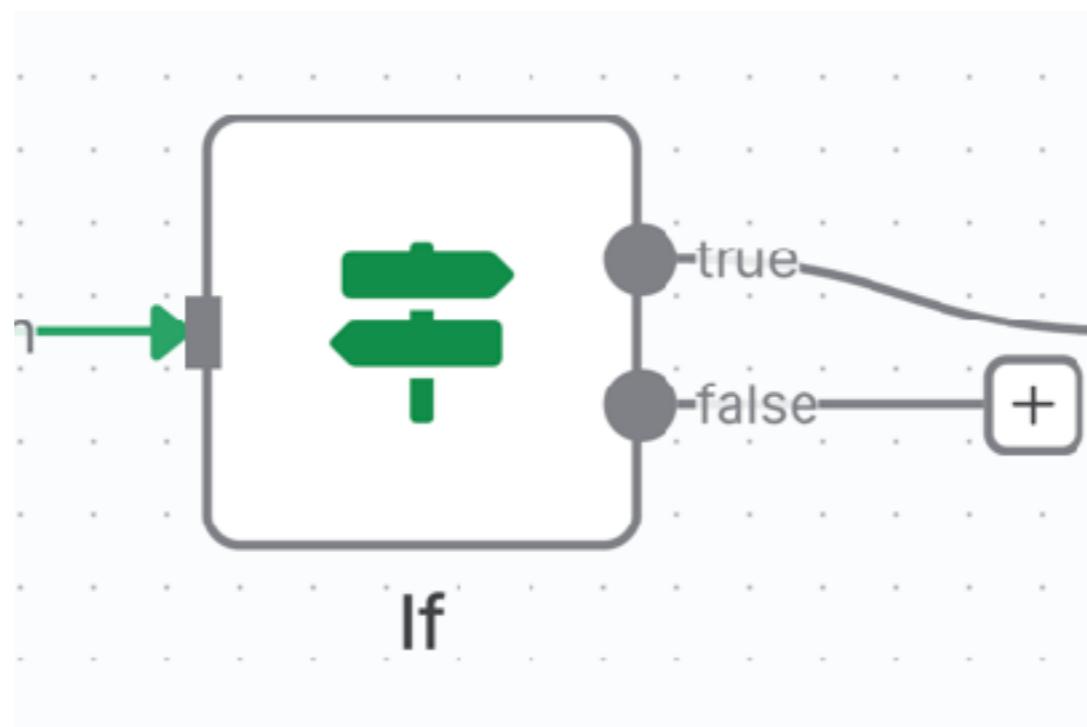


# Check type of event ?



# If node

Split workflow conditionally based on comparison

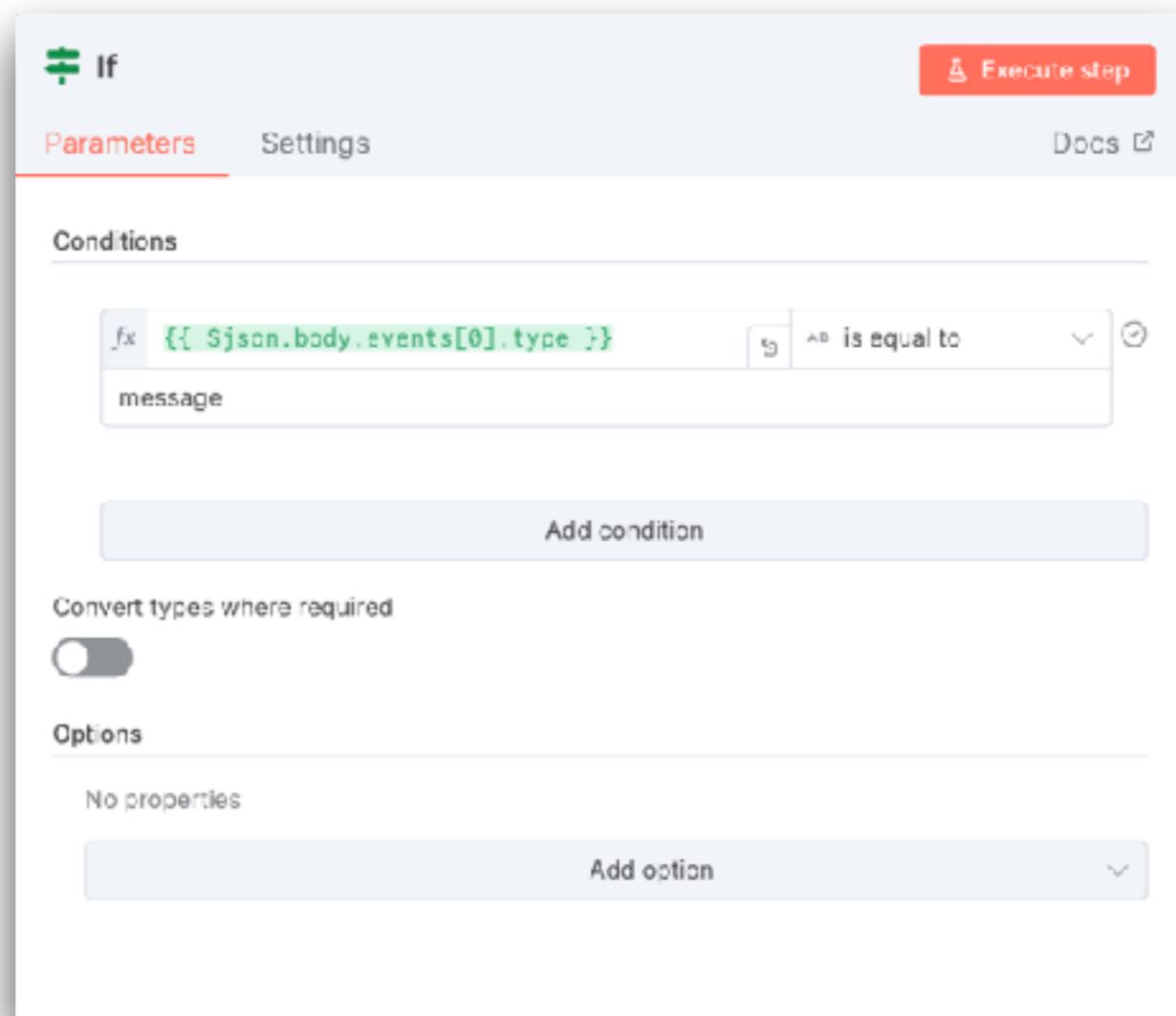


<https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.if/>



# Comparison operations

**Event.type is equal to message**



<https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.if/>



# Learning by doing

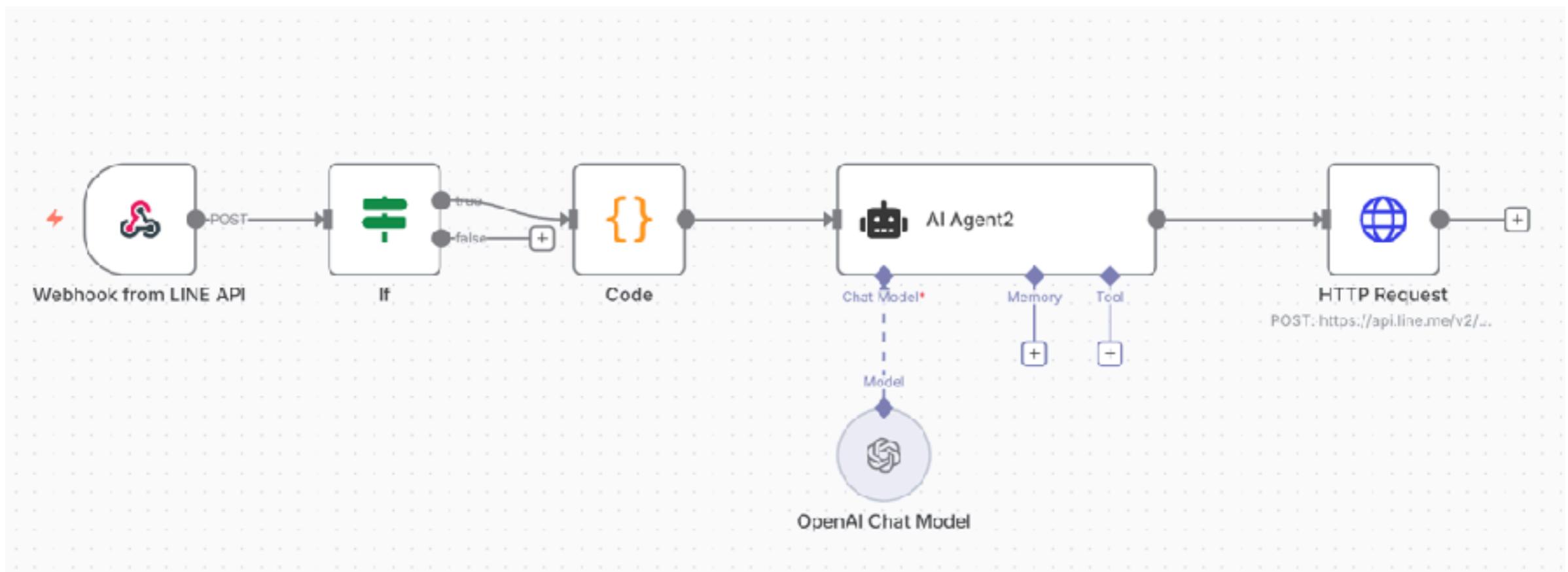


# ສືເໜືອງ ?





# Workflow in n8n



# Next workflow



# **RAG**

# **(Retrieval Augmented Generation)**



# RAG

Enhances LLMs by retrieving external knowledge before generating response

Improve accuracy

Reduce hallucinations

Real-time knowledge updates



# Core Components

## Retriever

Fetch relevant documents from a knowledge base

## Generator

Uses the retrieved information to generate a response

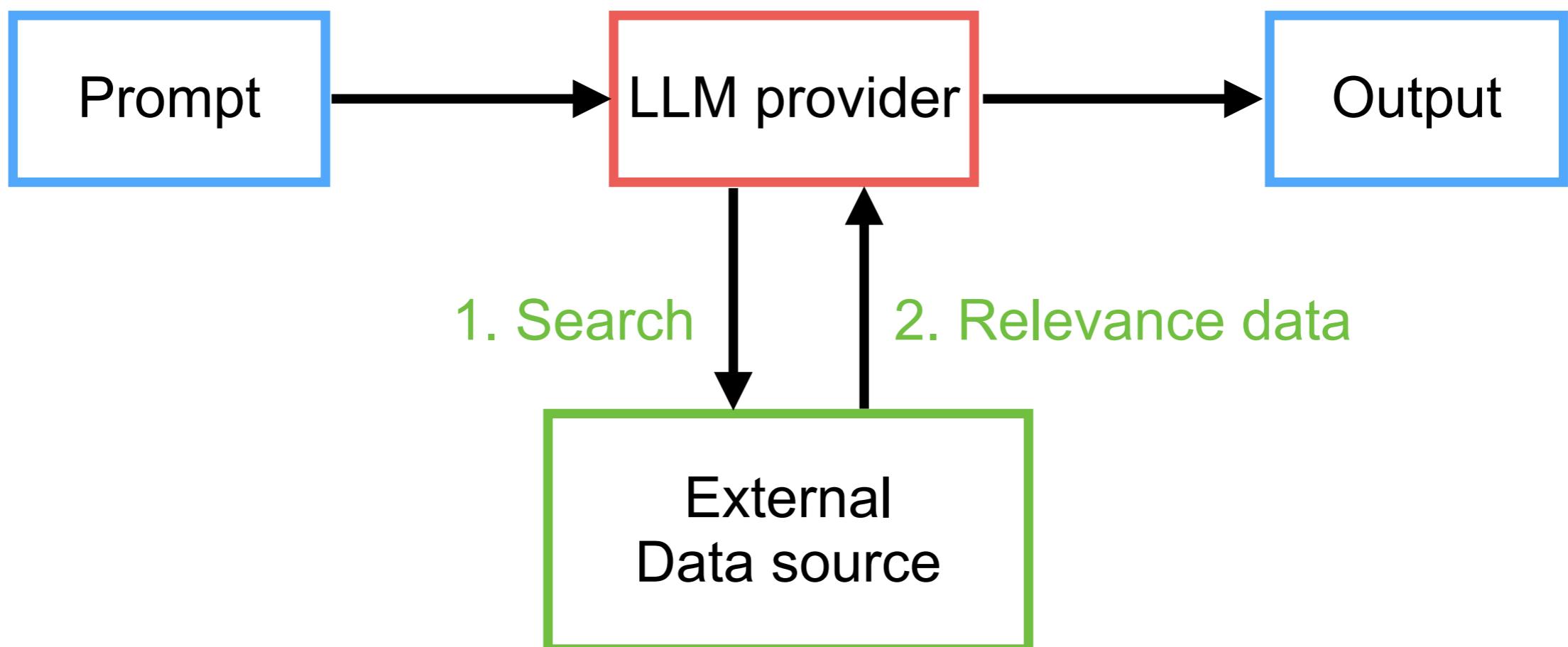
## Enhancements

Different RAG architectures modify how retrieval and generation interact to improve performance



# RAG ?

Fixed-knowledge !!



# How to search/retrieve data ?

Data source

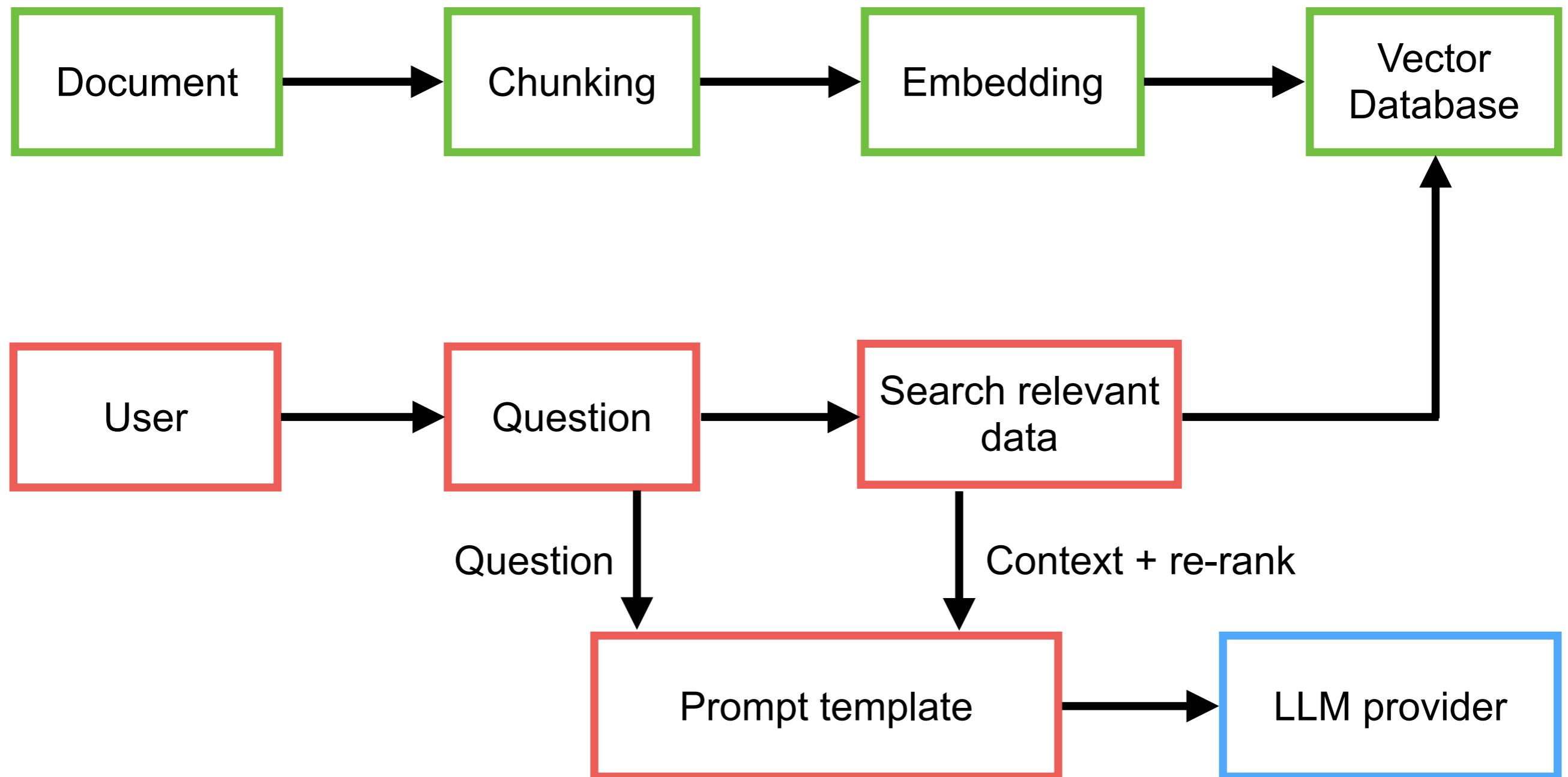
Data preparation

Search

Search result



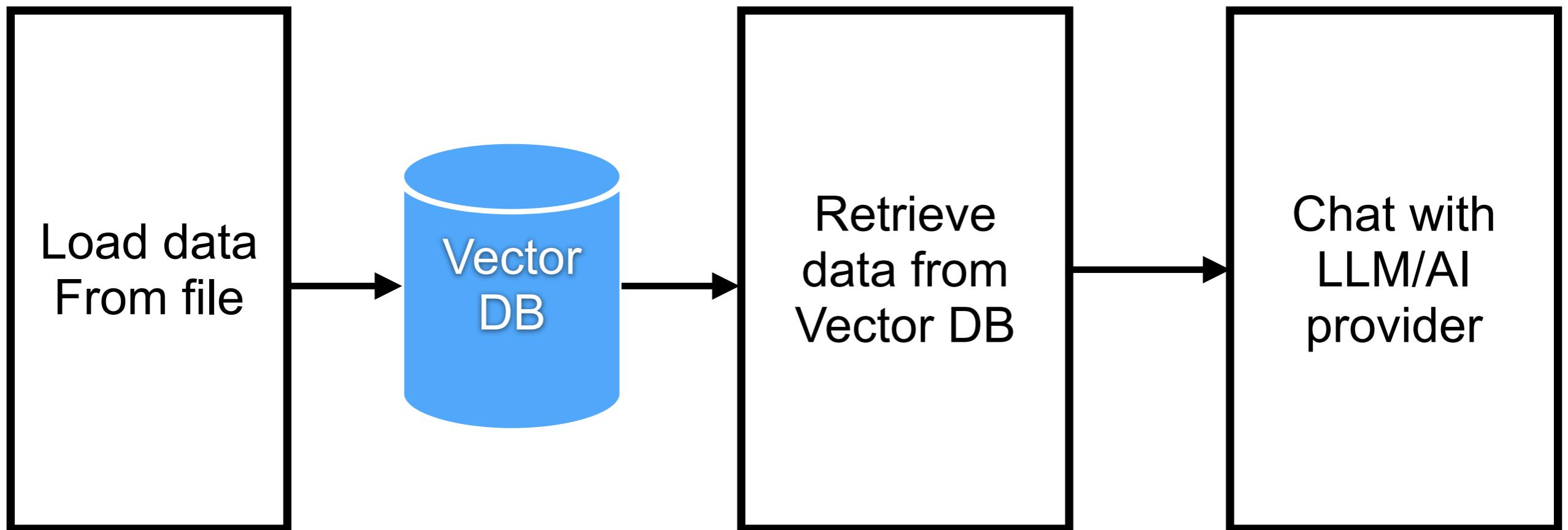
# RAG process



# Let's go with n8n



# RAG Flow



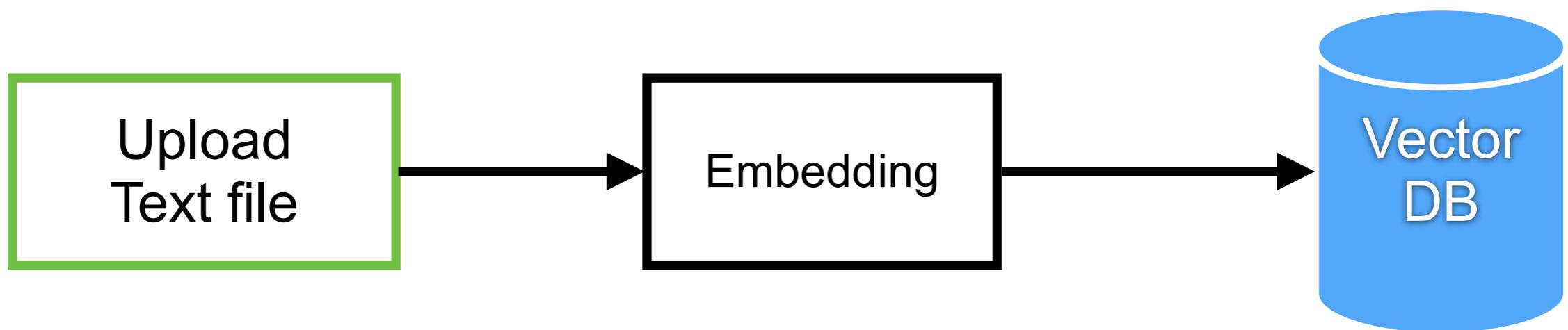
<https://docs.n8n.io/advanced-ai/rag-in-n8n/>



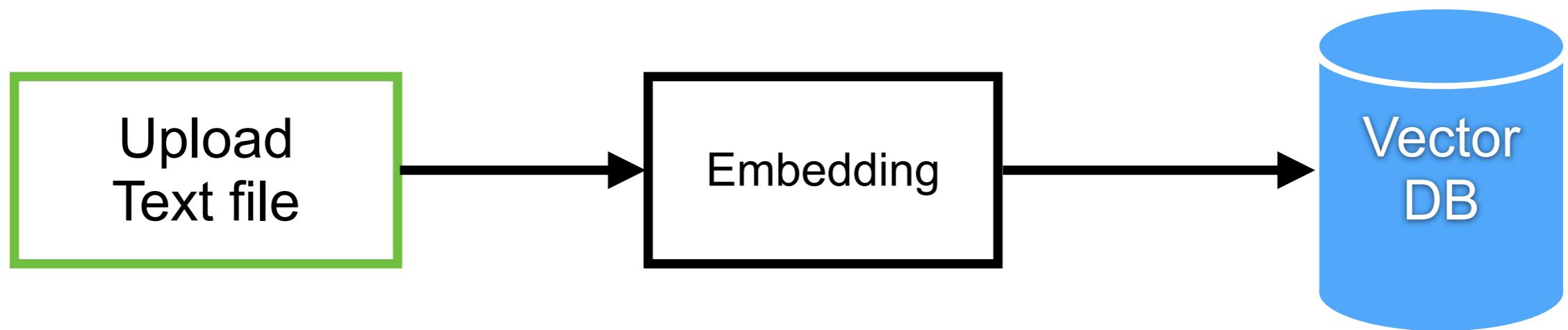
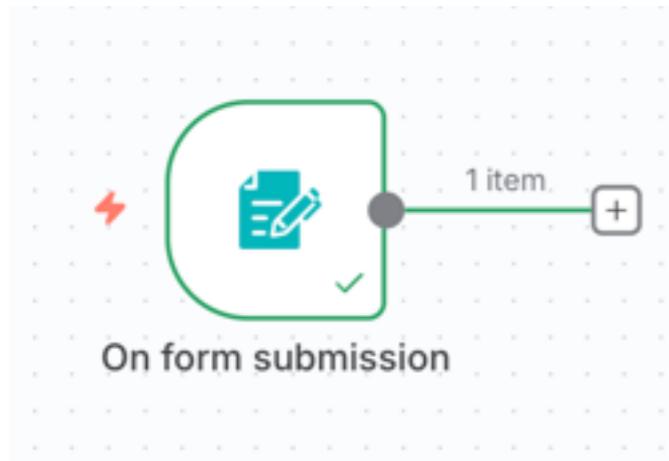
# Load data from file and store



# Load data from file and store



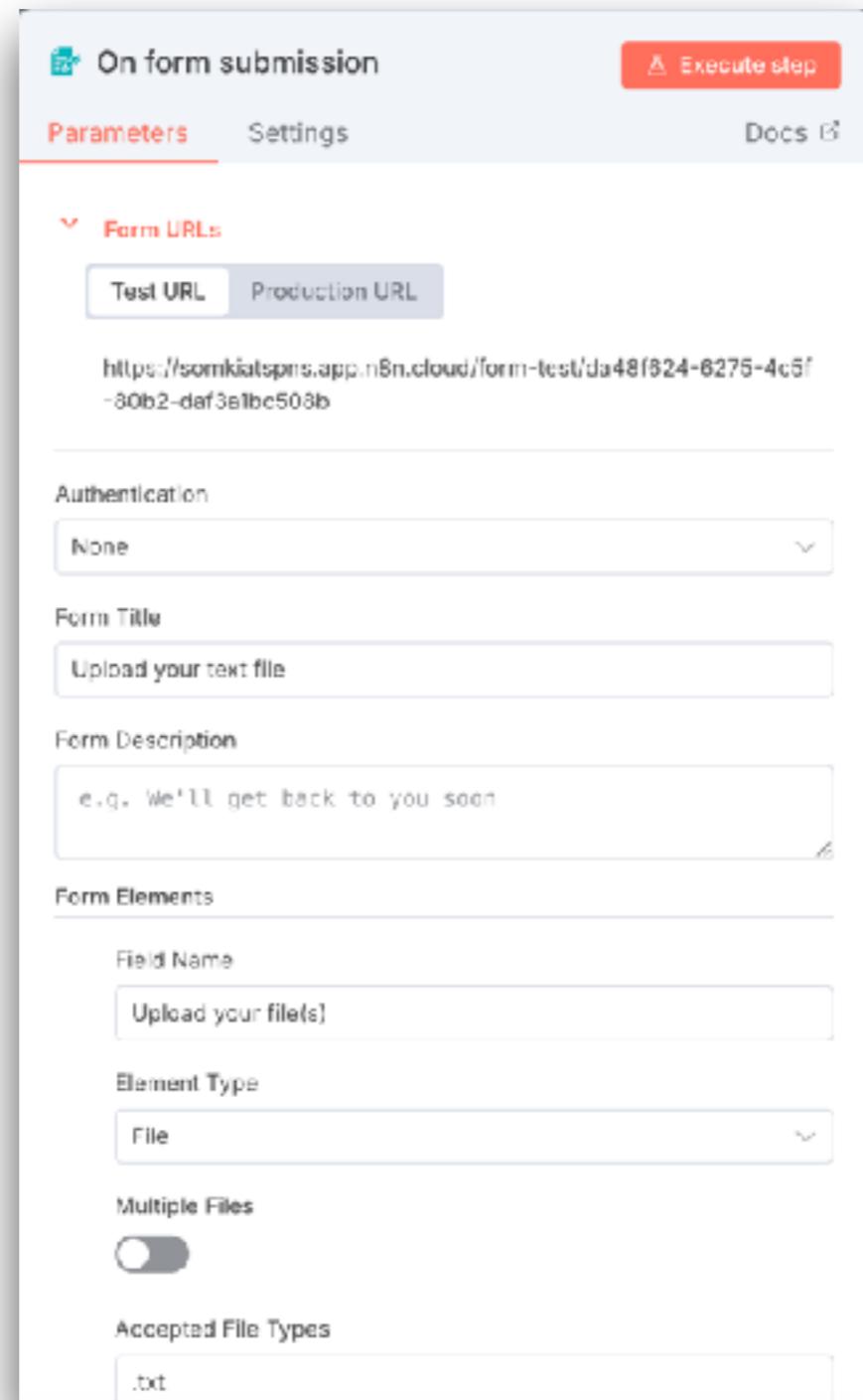
# 1. User upload file (Trigger)



<https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.formtrigger/>



# n8n Form Trigger Node



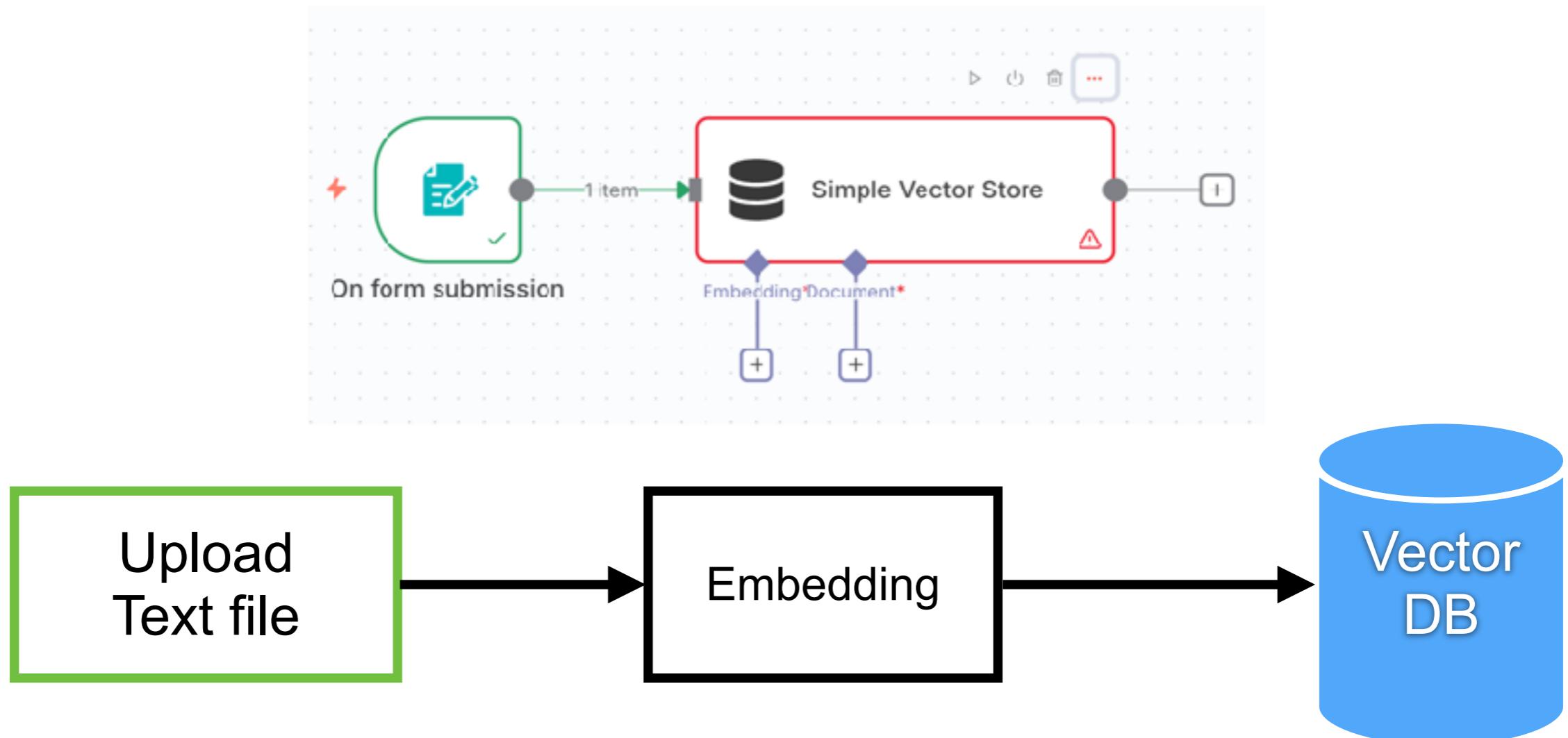
**Element type = File  
File type = .txt**

<https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.formtrigger/>



## 2. Embedding and Save to Vector DB

Use Simple Vector Store (experiment)



<https://docs.n8n.io/integrations/builtin/cluster-nodes/root-nodes/n8n-nodes-langchain.vectorstoreinmemory/>



# Simple Vector Store node

Store and retrieve embedding in n8n's memory

Simple Vector Store

Parameters Settings Does

Tip: Get a feel for vector stores in n8n with our [RAG starter template](#)

Operation Mode: Insert Documents

Memory Key: From list

Embedding Batch Size: 200

Clear Store:

For experimental use only: Data is stored in memory and will be lost if n8n restarts. Data may also be cleared if available memory gets low, and is accessible to all users of this instance. [More info](#)

Embedding:  Session, subSession, Logs

Document:

Embedding model

Store your documents

<https://docs.n8n.io/integrations/builtin/cluster-nodes/root-nodes/n8n-nodes-langchain.vectorstoreinmemory/>



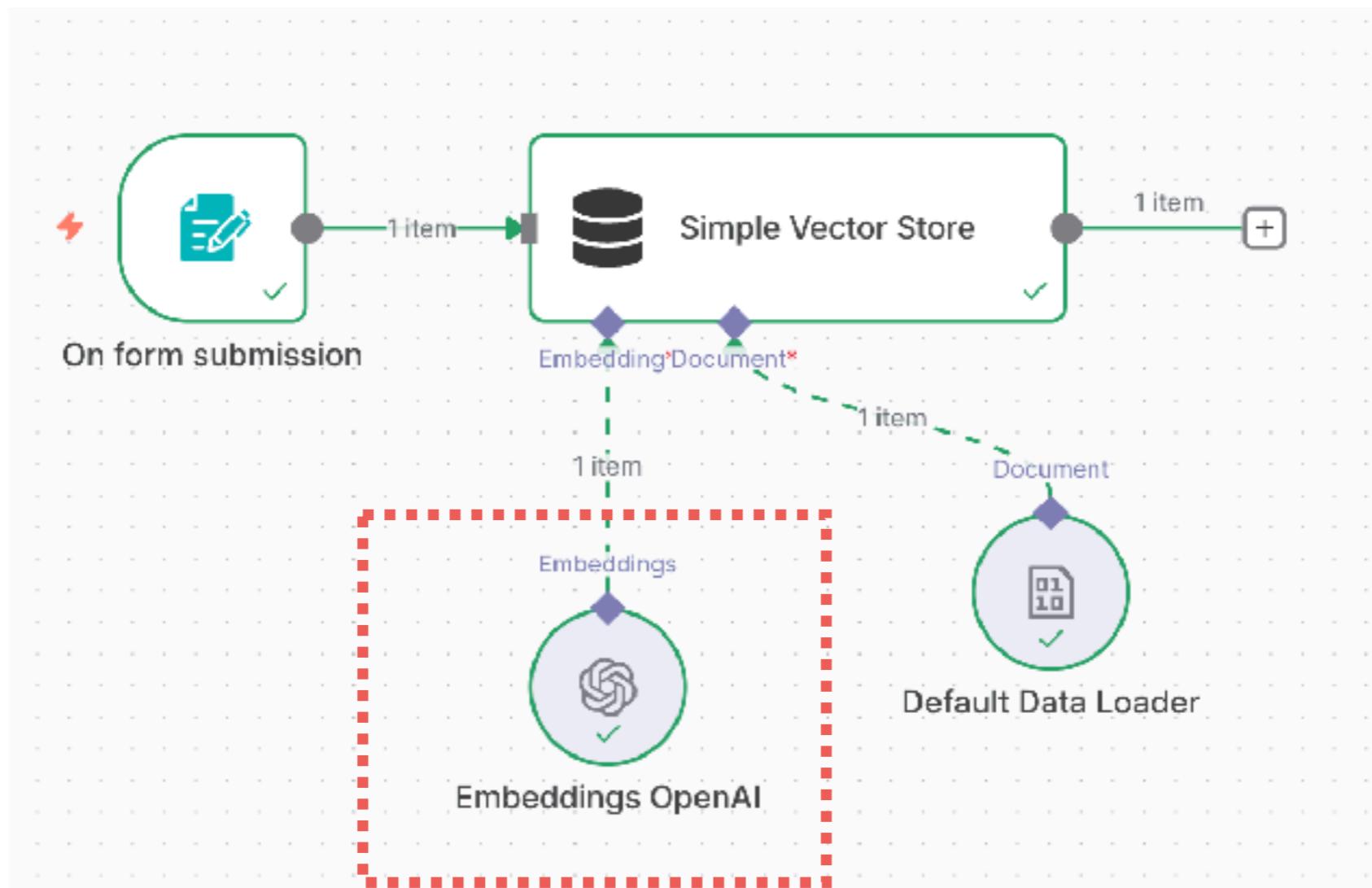
# Vector DB



Weaviate

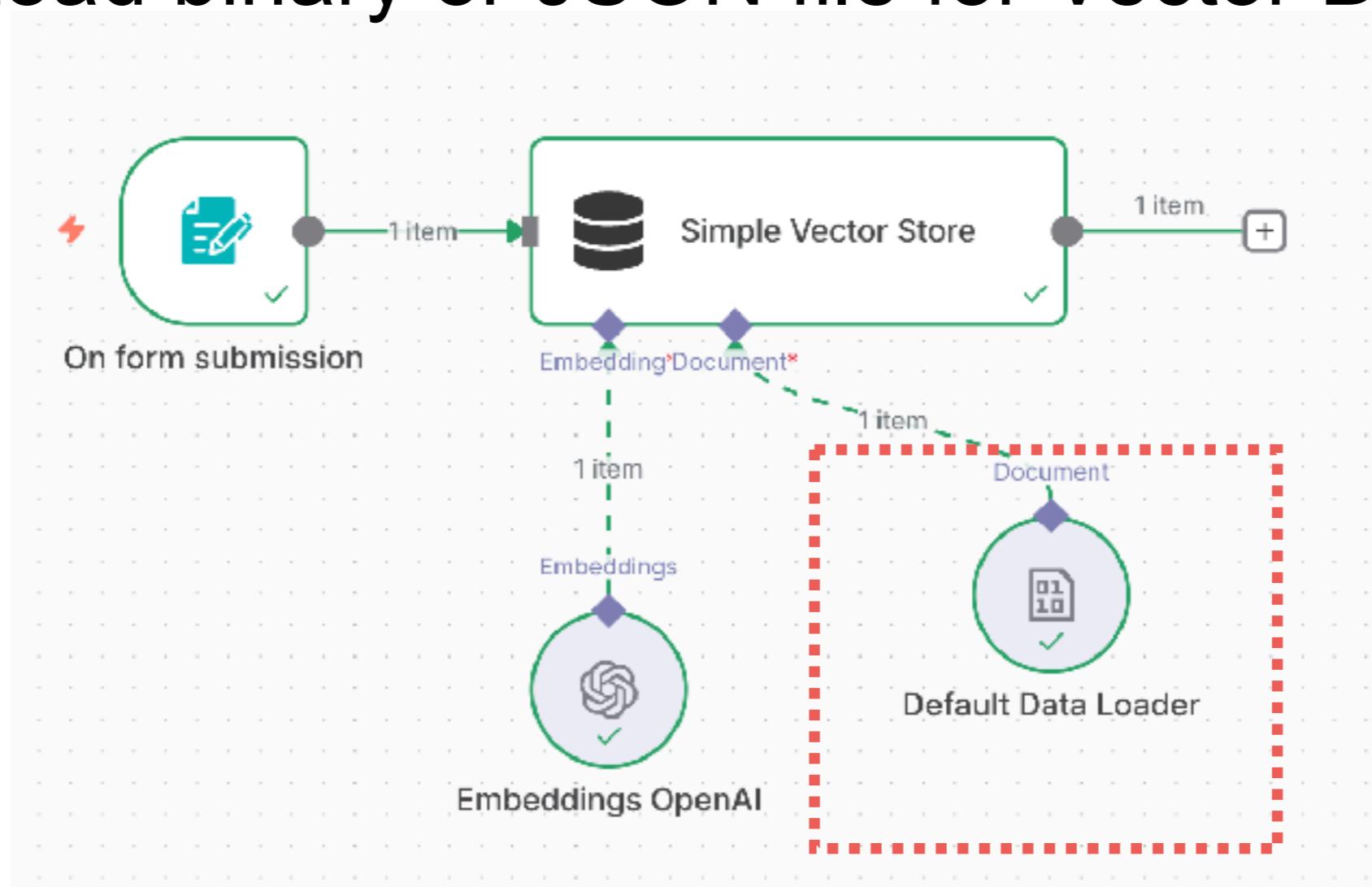


# Add embedding model



# Add Default Data Loader node

Load binary or JSON file for Vector DB



<https://docs.n8n.io/integrations/builtin/cluster-nodes/sub-nodes/n8n-nodes-langchain.documentdefaultdataloader/>



# Default Data Loader node

This will load data from a previous step in the workflow. [Example](#)

Type of Data

Binary

Mode

Load All Input Data

Data Format

Automatically Detect by Mime Type

Text Splitting

Simple

Splitting or Chunking

No properties

Add Option

<https://docs.n8n.io/integrations/builtin/cluster-nodes/sub-nodes/n8n-nodes-langchain.documentdefaultdataloader/>



# Testing

The screenshot shows the n8n interface with a "Simple Vector Store" node selected. The node has the following configuration:

- Operation Mode:** Insert Documents
- Memory Key:** From list, vector\_store\_key
- Embedding Batch Size:** 200
- Clear Store:** Off

A tip message at the top says: "Tip: Get a feel for vector stores in n8n with our [RAG starter template](#)". A warning message below states: "For experimental use only: Data is stored in memory and will be lost if n8n restarts. Data may also be cleared if available memory gets low, and is accessible to all users of this instance. [More Info](#)".

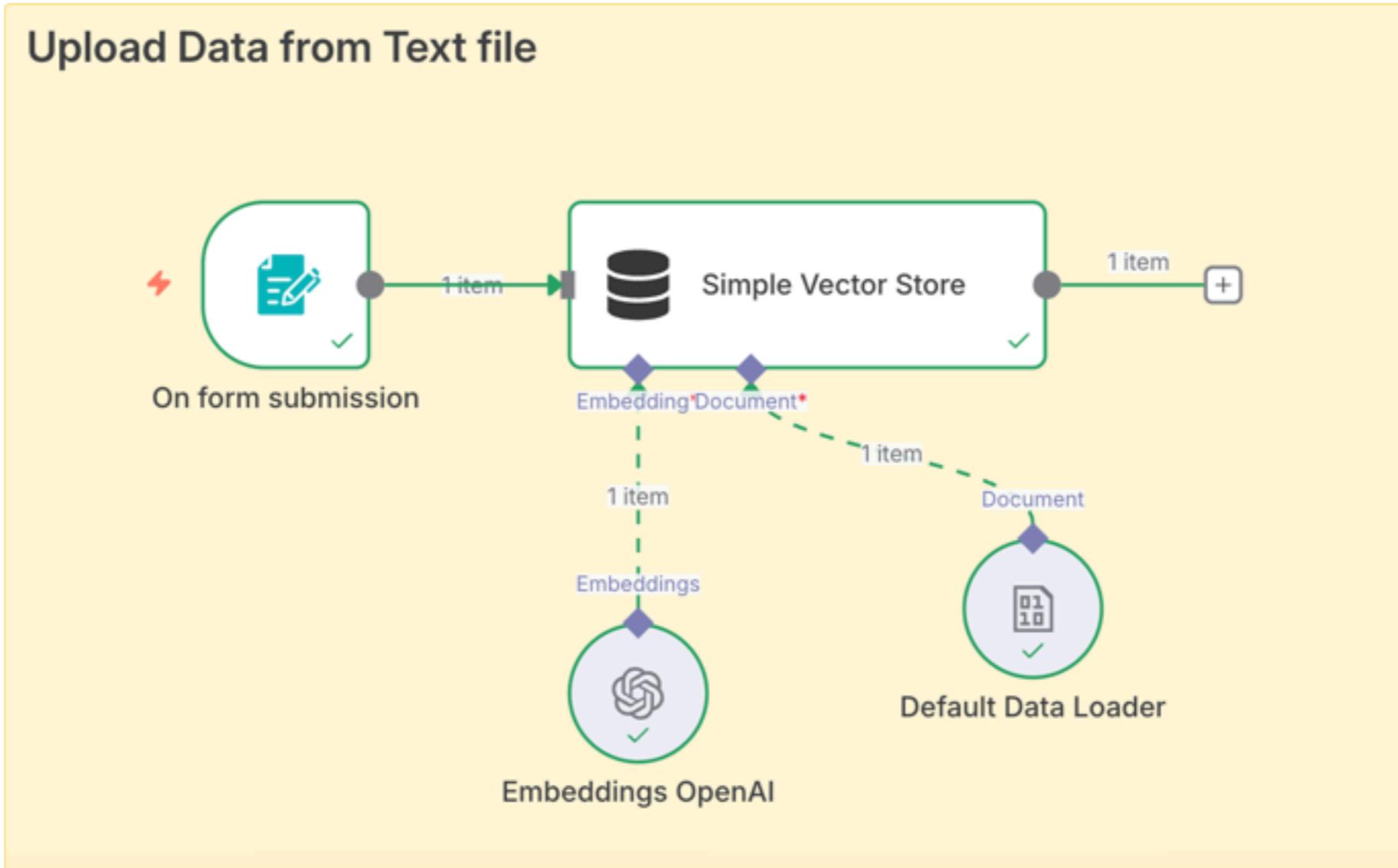
The "Output" tab shows the resulting data structure:

```
1 item
└─ metadata
    └─ source blob
    └─ blobType text/plain
    └─ doc
        └─ lines
            # from 1
            # to 76
            └─ pageContent
                Context : joke
                Content : A: 1
                Q: Why is it that when I wake up in the morning, I always feel like I'm falling?
                A: Because you're falling.
                Q: What's the best way to make a sandwich?
                A: Use bread.
                Q: What's the best way to make a sandwich?
                A: Use bread.
                Q: What's the best way to make a sandwich?
                A: Use bread.
                Q: What's the best way to make a sandwich?
                A: Use bread.
                Q: What's the best way to make a sandwich?
                A: Use bread.
                Q: What's the best way to make a sandwich?
                A: Use bread.
                Q: What's the best way to make a sandwich?
                A: Use bread.
                Q: What's the best way to make a sandwich?
                A: Use bread.
                Q: What's the best way to make a sandwich?
                A: Use bread.
                Q: What's the best way to make a sandwich?
                A: Use bread.
                Q: What's the best way to make a sandwich?
                A: Use bread.
                Q: What's the best way to make a sandwich?
                A: Use bread.
                Q: What's the best way to make a sandwich?
                A: Use bread.
                Q: What's the best way to make a sandwich?
                A: Use bread.
                Q: What's the best way to make a sandwich?
                A: Use bread.
                Q: What's the best way to make a sandwich?
                A: Use bread.
```

The "Logs" tab shows the following message: "I wish this node would..."



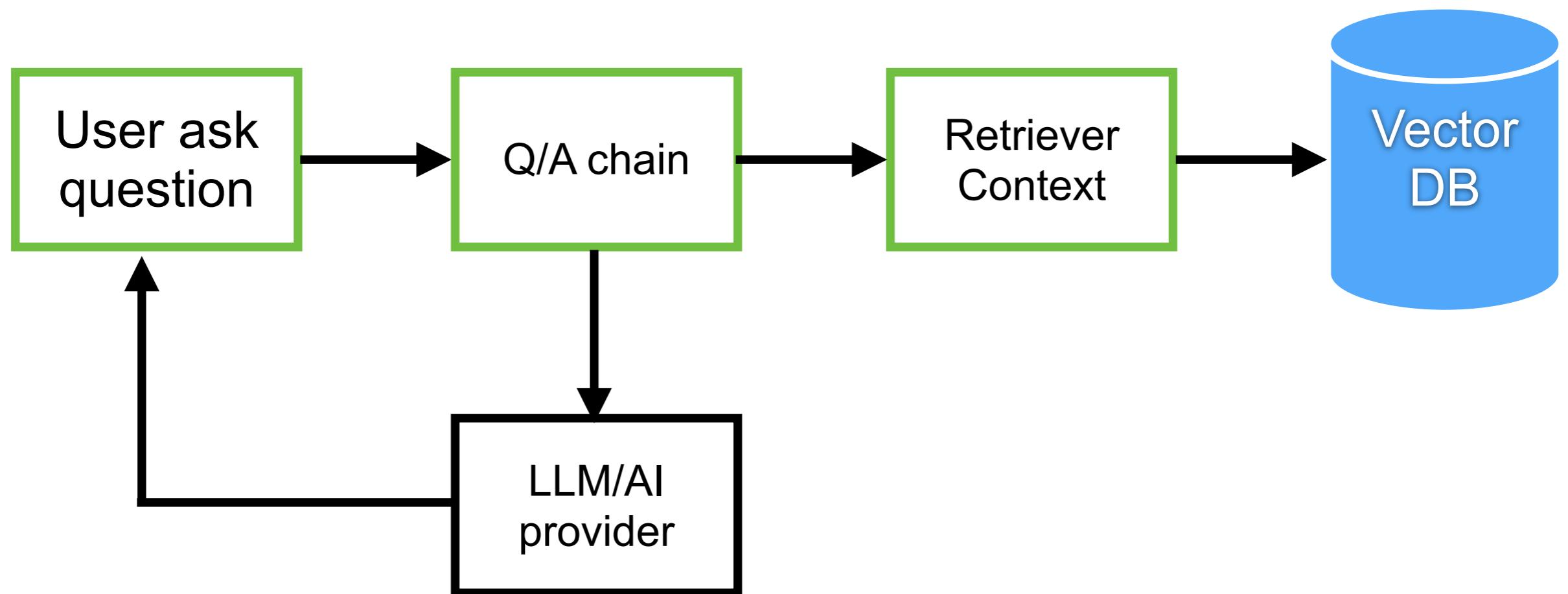
# Done with Step 1 !!



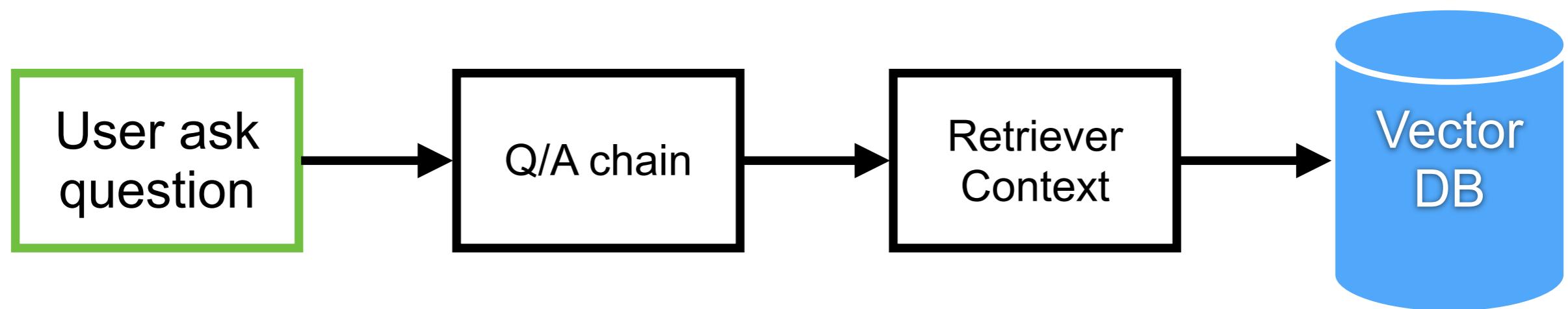
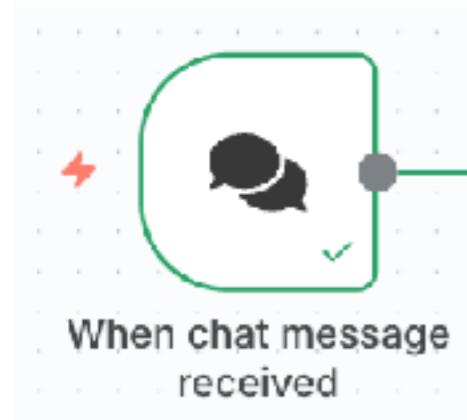
# **Retrieve data from Vector DB and Chat with AI**



# 1. Retrieve data from Vector DB



# Add Chat trigger node

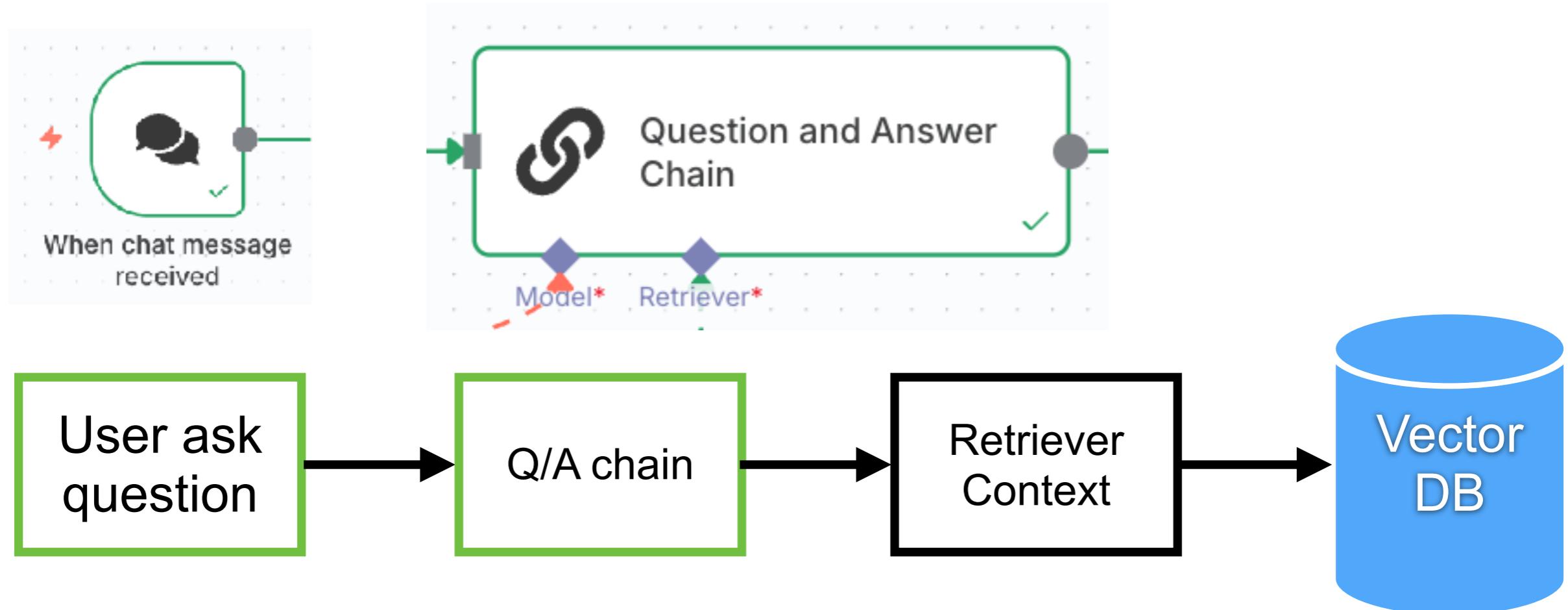


<https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-langchain.chattrigger/>



# Add AI Agent Node

Use Q/A chain from Langchain



<https://docs.n8n.io/integrations/builtin/cluster-nodes/root-nodes/n8n-nodes-langchain.agent/conversational-agent/>



# Use Q/A chain from Langchain

Retrieve data from Vector DB  
Choose LLM/AI model

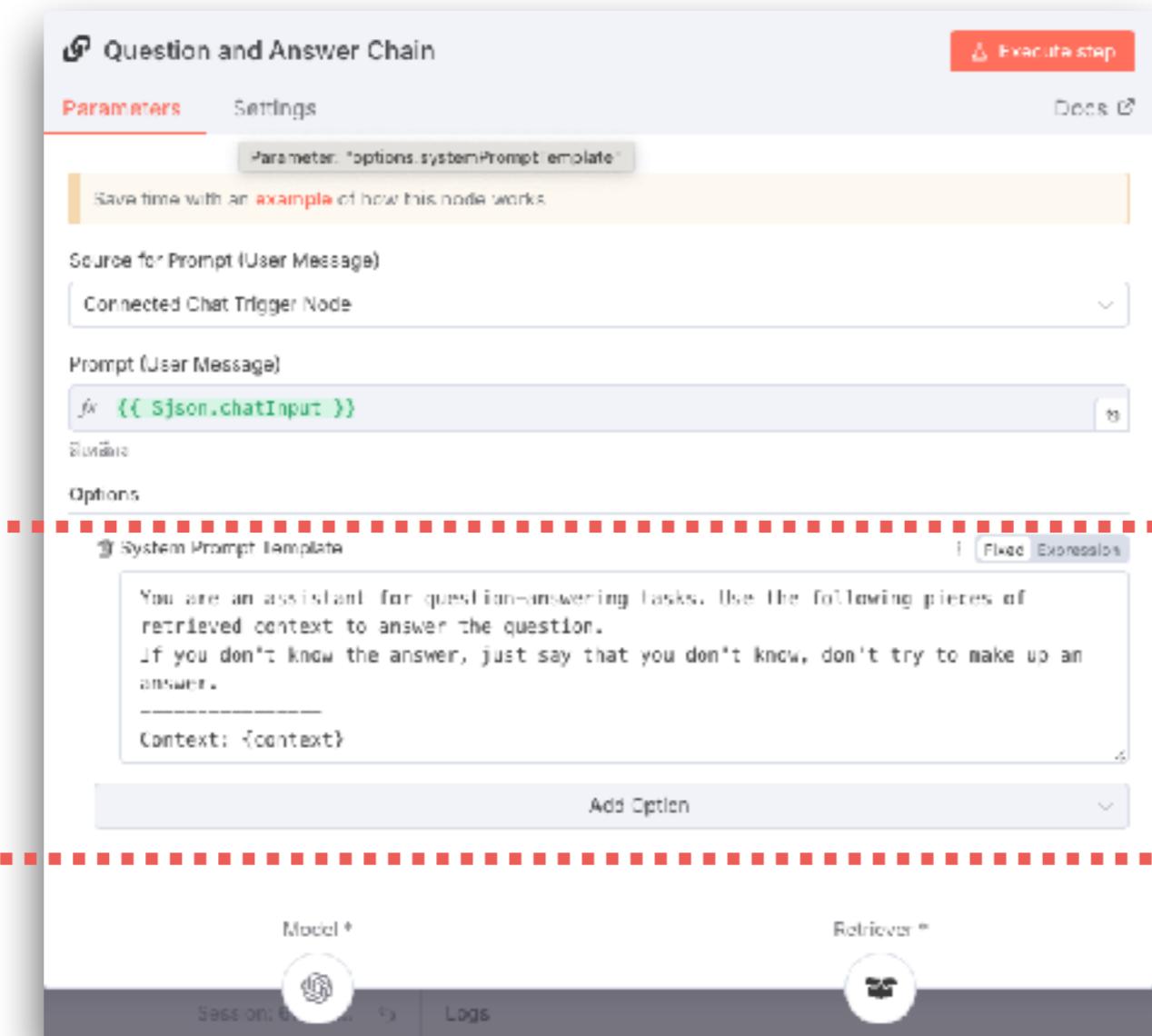


<https://docs.n8n.io/integrations/builtin/cluster-nodes/root-nodes/n8n-nodes-langchain.agent/conversational-agent/>



# Use Q/A chain from Langchain

## Provider System prompt template

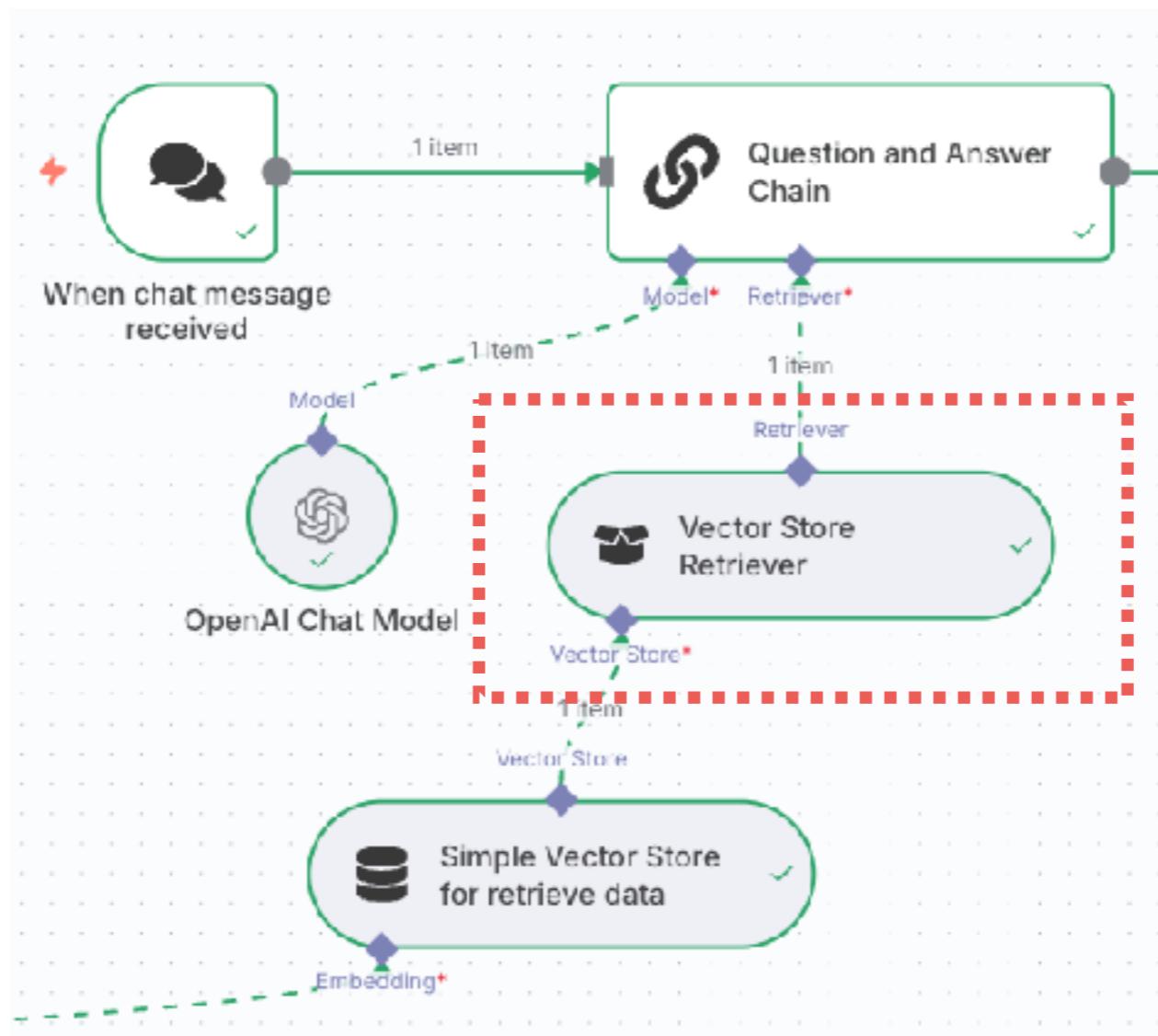


<https://docs.n8n.io/integrations/builtin/cluster-nodes/root-nodes/n8n-nodes-langchain.agent/conversational-agent/>



# Add vector store retriever node

Retrieve data from Vector DB

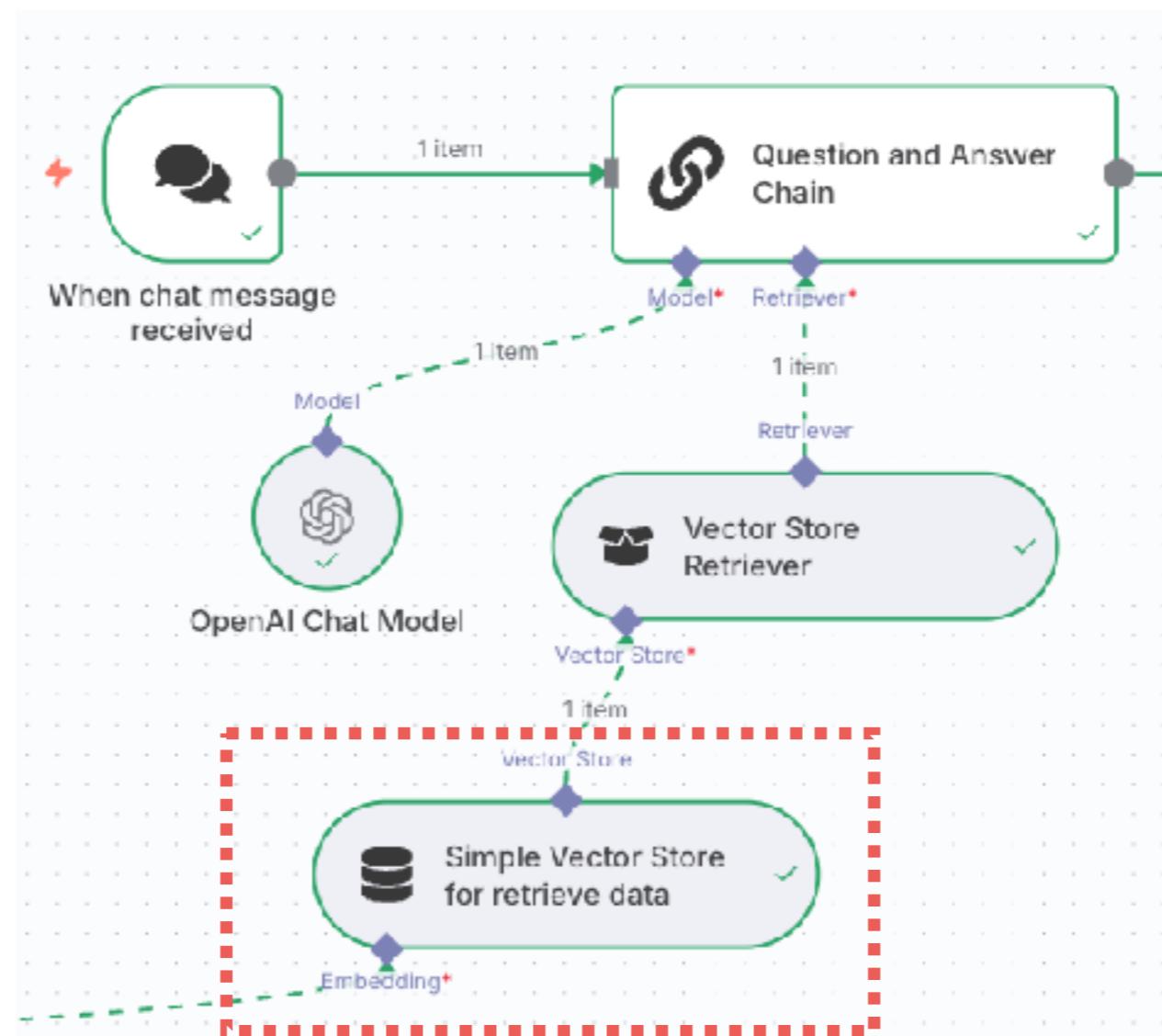


<https://docs.n8n.io/integrations/builtin/cluster-nodes/sub-nodes/n8n-nodes-langchain.retrievervectorstore/>



# Connect to Simple Vector Store

Retrieve data from Vector DB (memory)



<https://docs.n8n.io/integrations/builtin/cluster-nodes/root-nodes/n8n-nodes-langchain.vectorstoreinmemory/>



# Connect to Simple Vector Store

Retrieve data from Vector DB (memory)

Simple Vector Store for retrieve data

Parameters    Settings    Docs

Tip: Get a fee for vector stores in n8n with our [RAG starter template](#)

Operation Mode: Retrieve Documents (As Vector Store for Chain/Tool)

Memory Key: From list: vector\_store\_key

Rerank Results:

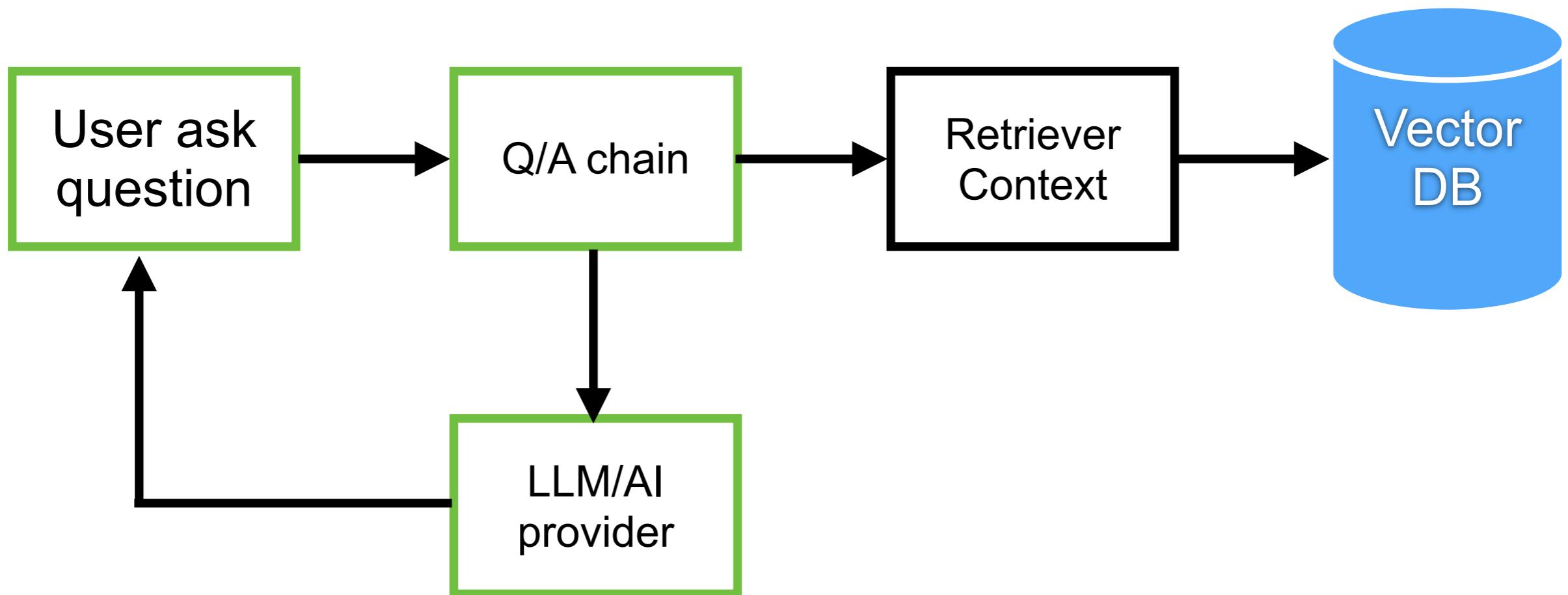
For experimental use only: Data is stored in memory and will be lost if n8n restarts. Data may also be cleared if available memory gets low, and is accessible to all users of this instance. [More info](#)

**Retrieve mode**

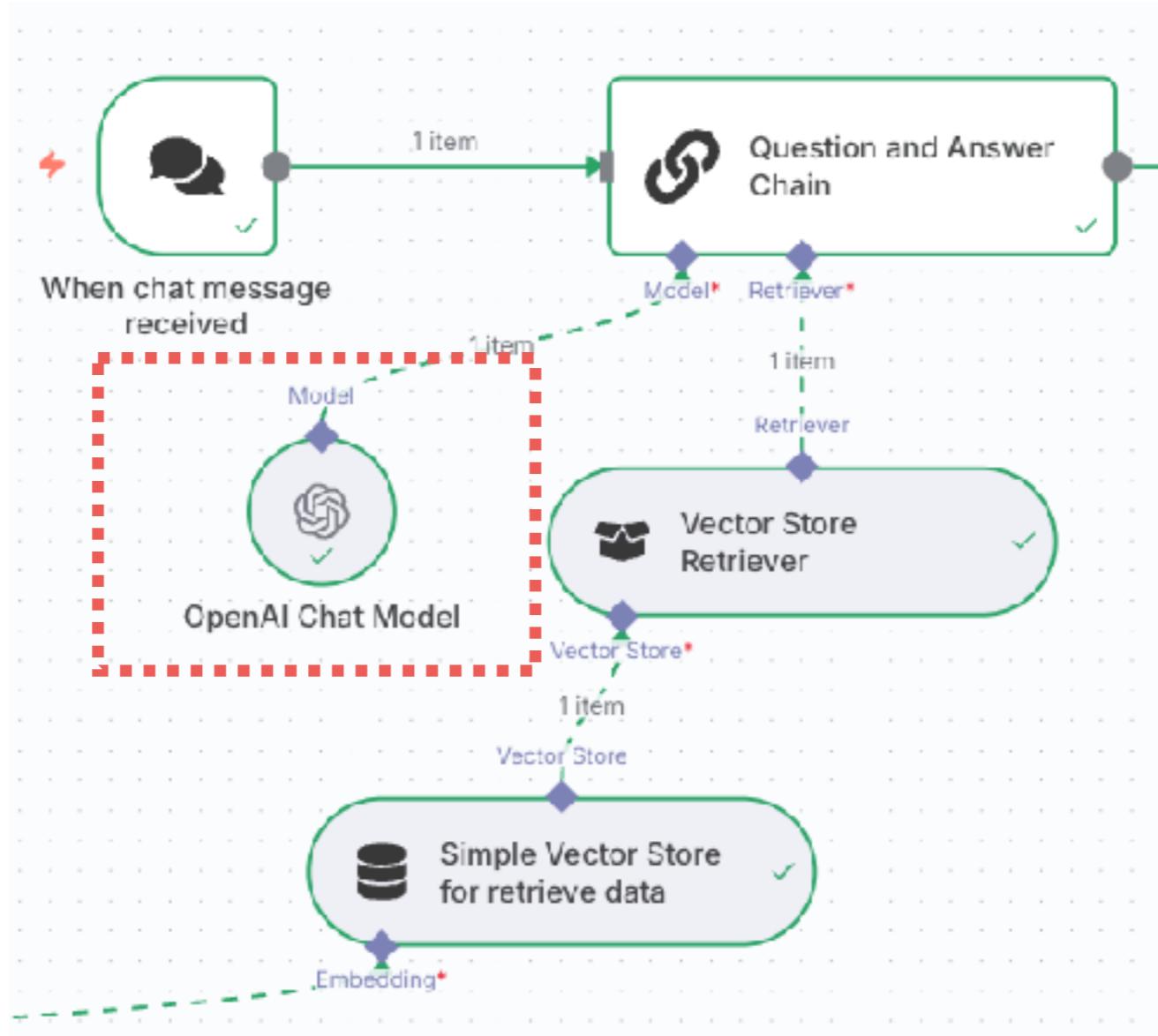
<https://docs.n8n.io/integrations/builtin/cluster-nodes/root-nodes/n8n-nodes-langchain.vectorstoreinmemory/>



## 2. Send question and context



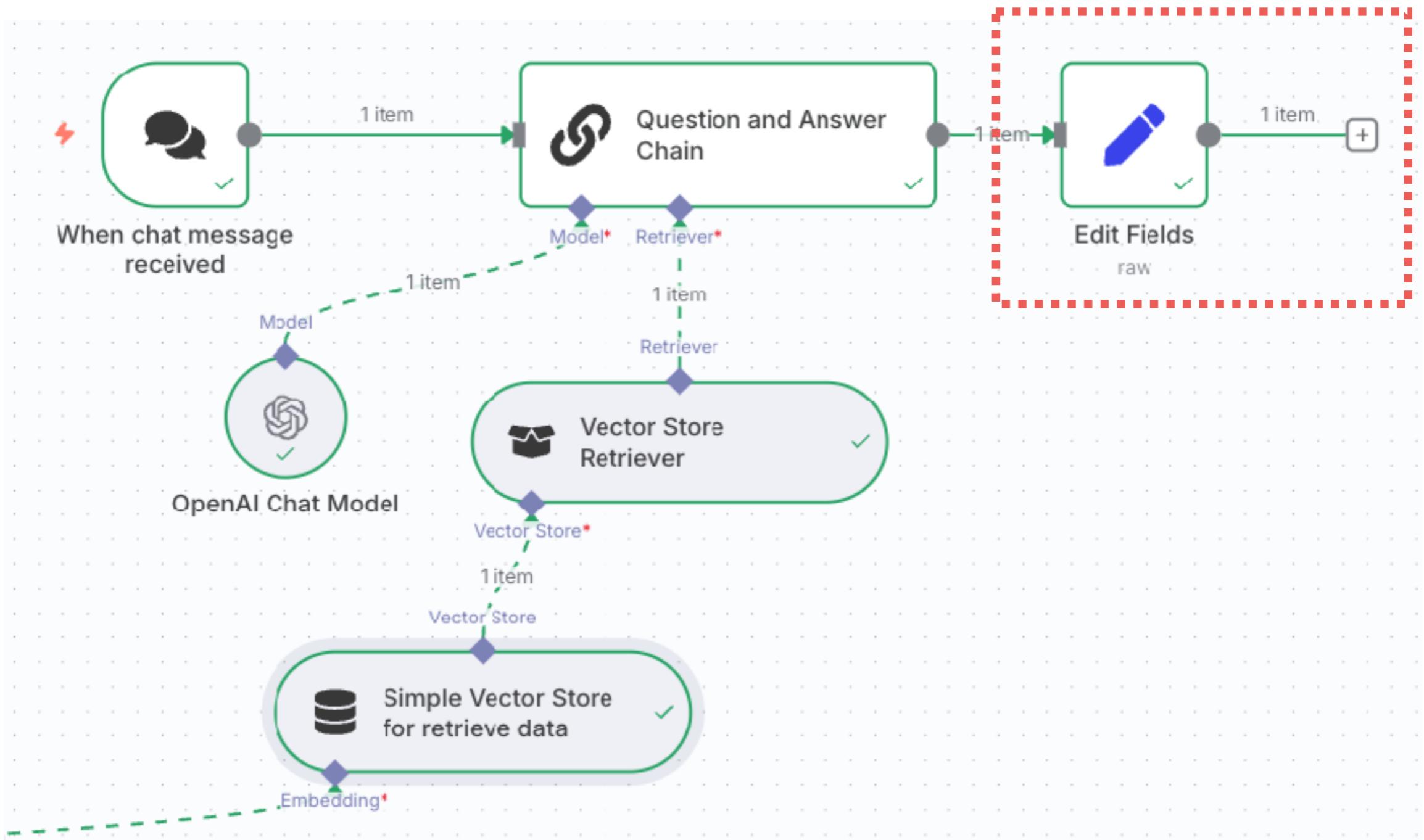
# Add OpenAI Chat Model



<https://docs.n8n.io/integrations/builtin/cluster-nodes/root-nodes/n8n-nodes-langchain.vectorstoreinmemory/>



# Try to change output response



<https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.set/>



# Next workflow

