

Your first AI node: Basic LLM Chain

You will build an AI Email Classifier – a workflow that reads incoming emails and automatically labels them according to intent. No programming code is needed. Just a prompt, an LLM, and data flow patterns you already know.

In Lesson 2, you learned about how data flows through nodes and expressions . Now, we're going to create a real node that can think. You'll build an AI Email Classifier—a workflow that reads incoming emails and automatically labels them purposefully. No programming code is needed. Just a prompt, an LLM , and the data flow patterns you already know.

Classification of AI nodes in n8n

n8n has two types of AI nodes, and understanding the difference will help you avoid using a cannon when a slingshot would also work.

Root node - These are independent nodes that you drag onto the canvas. They perform AI tasks:

1. **Basic LLM Chain** - Send a prompt, receive a response. No tools, no memory. (This lesson)
2. **AI Agent** - An automated agent with tools, memory, and multi-step reasoning capabilities. (Lesson 4)
3. **Q&A Chain** - Answering questions from the document. (Lesson 6)
4. **Summarization Chain** - Summarizing long texts.
5. **Text Classifier** - Classifies text into categories.
6. **Sentiment Analysis** - Identifying positive/negative/neutral emotions.

Child nodes - These are attached to root nodes to extend their functionality:

1. LLM providers - OpenAI, Claude, Gemini , Groq, Ollama (you choose whichever model powers the root node)
2. Memory Management - Simple, PostgreSQL, Redis (Lesson 5)
3. Tools - SerpAPI, Wikipedia, Code, HTTP Request (Lesson 4)
4. Vector storage - Supabase, Pinecone, Qdrant (Lesson 6)
5. Embedded - OpenAI, Cohere, local models

The root node defines the behavior. The child nodes define the capabilities. An AI Agent root node with an OpenAI child node and a SerpAPI engine child node creates an agent that uses GPT-40 and can search the web.

? **Quick test** : You need to summarize a long document. Which root node would you choose - Basic LLM Chain, AI Agent , or Summarization Chain?

Answer : Summarization Chain. It's specifically designed for summarization with optimal segmentation capabilities. Basic LLM Chain can also work, but you need to handle long documents manually. AI Agent is overkill – you don't need the tools or memory for simple summarization.

Building: An AI Email Sorting Tool

Build your first AI workflow. This classification tool reads an email and categorizes it into one of the following types: Request, support, urgent, or spam.

Step 1: Set up the trigger (Test mode)

Start a new workflow. Instead of Gmail Trigger (which requires a real email address), use this test setup:

1. Add a **Manual Trigger node**.
2. Add the **Set** node and create the following fields:
 1. subject(string): "Server down – production is broken"
 2. from(string): "ops-team@company.com"
 3. body(string): "Our production server went down at 3am. All customer-facing services are offline. Need immediate help".

This simulates an incoming email. Once the workflow is ready, you'll replace this with a real Gmail Trigger.

Step 2: Add Basic LLM Chain

1. Add a **Basic LLM Chain** node after the Set node.
2. Click on the node to open its configuration.
3. Under the **Model** section, click to add an **OpenAI Chat Model child node**.
4. Choose your OpenAI credentials.
5. Model: gpt-4o-mini(fast, inexpensive, smart enough for sorting)
6. **In the prompt** field, write:

Phân loại email này vào ứng một danh mục. Danh mục: - inquiry: các câu hỏi
i v? s?n ph?m, d?ch v? ho?c giá c? - support: các v?n ?? k? thu?t, l?i ho?
c yêu c?u tr? giúp - urgent: các v?n ?? c?n x? lý ngay l?p t?c - spam: th?
ti?p th? không mong mu?n ho?c tin nh?n không liên quan Tiêu ??
email: {{ \$json.subject }} Ng??i g?i email: {{ \$json.from }} N?
i dung email: {{ \$json.body }} Tr? l?i b?ng ứng m?t t?
: inquiry, support, urgent, spam Không c?n gi?i thích. Không c?n d?
u câu. Ch? c?n t? danh m?c.

Pay attention to the expressions - {{ \$json.subject }}, {{ \$json.from }}, {{ \$json.body }}
- which retrieve data from the output of the Set node. This is the connection between the data stream of Lesson 2
and the AI ??of Lesson 3.

Step 3: Routing based on classification

Add an IF node after the Basic LLM Chain. Configure it:

1. Condition: `{{ $json.text }}` contains the word `urgent`
2. True branch: Add a **Slack** node (or another notification node) to alert your team.
3. False branch: Connects to a **Google Sheets** node to record classifications.

The field `$json.text` contains the LLM response - in this case, a single word such as "urgent" or "support".

Step 4: Check

Click on "Test workflow". Observe the data flow:

1. Manual activation
2. Node Set creates fake email data.
3. Basic LLM Chain data classification (you will see "urgent" in the results table)
4. The IF node routes data to the appropriate branch.

Change the test data of the Set node to different email scenarios and run it again. Try a sales email (which will be classified as spam), a pricing inquiry, and a bug report (support).

? **Quick check** : What happens if the LLM returns "URGENT" (uppercase) but your IF node checks for "urgent" (lowercase)?

Answer : The condition will not be met. Fix this by normalizing the LLM output using a Set-- node expression `{{ $json.text.toLowerCase() }}` or by using "contains" instead of "equals" in the IF node with the comparison set to case-insensitive. Always normalize the LLM output before routing.

Techniques for creating prompts for n8n

Writing prompts for automation is different from chatting with an LLM. In a conversation, detailed feedback is sufficient. In a workflow, you need predictable, analyzable output that subsequent nodes can process.

Three rules for prompt n8n:

1. Clearly define the output format limitations.

Sai: "Email này thu?c lo?i nào?" ?úng: "Tr? l?i ch? b?ng m?t t?
: inquiry, support, urgent, spam"

2. Provide complete context in the prompt. LLMs don't store information between executions. Each prompt must include all the necessary data—don't assume it "knows" what you're working with.

3. Use examples (few-shots) for complex tasks.

Ví d?: - "B?n có th? cho tôi bi?t v? giá c? không?" ? inquiry - "Nút xu?t d
? li?u không ho?t ??ng" ? support - "C? s? d? li?u s?n xu?t b? l?i" ?
urgent - "Mua ??ng h? gi?m giá ngay!" ? spam Bây gi? hãy phân lo?
i: `{{ $json.body }}`

Few-shot examples significantly improve classification accuracy. For tasks where the boundaries between categories are unclear (e.g., is "our report is late" urgent or does it require support?), adding 3–5 examples per category can make the difference between 70% and 95% accuracy.

From testing to production

After the classifier has worked with the test data, change the trigger:

1. Remove or disconnect the Manual Trigger and Set node.
2. **Add the Gmail Trigger** node at the beginning.
3. Configure it with your Gmail login information.
4. Set the trigger to "New Email"
5. Update the expressions in the Basic LLM Chain to match the output structure of Gmail:
 1. Subject: `{{ $json.subject }}`(same)
 2. From: `{{ $json.from.value[0].address }}`
 3. Body: `{{ $json.text }}`(or `{{ $json.snippet }}` for shorter text)

Now, activate the workflow. Every new email that arrives will be automatically categorized and routed.

Key points to remember

1. The root nodes (Basic LLM Chain, AI Agent, Q&A Chain) define the AI's behavior; the child nodes (LLM provider, memory, tools) define its capabilities.
2. Basic LLM Chain is the optimal choice for simple query-response tasks: classification, summarization, and extraction.
3. Question output limits - a predictable format that ensures reliable next-line routing.
4. Use the Manual Trigger + Set node for development, then switch to using the actual trigger for the production environment.
5. Few-shot examples in the prompt significantly improve classification accuracy.

1. Question 1:

You're building an email classification tool and want to test it without waiting for actual emails. What's the best approach?

1. A. Send a test email to yourself and wait for Gmail to activate.
2. B. Replace the Gmail Trigger with a Manual Trigger and use the Set node to simulate email data.
3. C. Disable the workflow and manually paste the data into each node.

EXPLAIN:

This is a core n8n development template. Use the Manual Trigger + Set node to generate dummy test data during development. Once the workflow is working, replace the Manual Trigger with a real Gmail Trigger. You can even keep both triggers connected to the same thread and disable the trigger you're not using.

2. Question 2:

Your email classification tool is returning 'Category is: support request' instead of just 'support'. How can I fix this?

1. A. Use a longer, more detailed prompt.
2. B. Add output constraints to your prompt - specify the exact format such as 'Reply with exactly one word: inquiry, support, urgent, spam'
3. C. Switch to a more robust model such as GPT-4o

EXPLAIN:

Default LLM models are very detailed. The solution lies in the prompt, not the model. Clearly define the output format: 'Reply with exactly one word from this list: inquiry, support, urgent, spam. No explanation. No punctuation.' This makes the output predictable and analyzable by subsequent nodes.

3. Question 3:

What is the difference between a Basic LLM Chain and an AI Agent node?

1. A. Basic LLM Chain is faster because it bypasses the LangChain framework.
2. B. Basic LLM Chains send a prompt and receive a response—no tools, no memory, no decision-making process. AI Agents can use tools, maintain memory, and make multi-step decisions.
3. C. There is no difference - they are aliases for the same node.

EXPLAIN:

Basic LLM Chain is a simple process that takes prompt input and receives feedback output. It's perfect when you only need an LLM for classifying, summarizing, or transforming text. The AI Agent (lesson 4) adds autonomy – it can call tools, remember conversations, and chain multiple inference steps. Use the simplest node that suits your task.

Submit your work

Training results

You have completed **0** questions.

-- / --

[Review the lesson](#)

You finished reading the article "**Your first AI node: Basic LLM Chain**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.