How To Build Your First AI Agent (+Free Workflow Template)

如何建立您的第一個 AI 代理人（附免費工作流程範本）

Step-by-step guide to building AI agents with three practical approaches—coding from scratch for full control, leveraging powerful frameworks like LangChain or LlamaIndex for faster development, or using no-code tools like n8n for rapid prototyping and automation.

建立 AI 代理人的逐步指南，包含三種實用方法——從零編碼以獲得完全控制，利用強大的框架如 LangChain 或 LlamaIndex 加速開發，或使用無需編碼的工具如 n8n 進行快速原型設計與自動化。

Mihai Farcas 米哈伊·法爾卡斯

April 24, 2025 · 12 minutes read

· 12 分鐘閱讀

Imagine building an assistant that can research topics online, summarize the findings, and save them directly to your Notion – automatically. That’s the kind of intelligent

automation AI agents make possible.

想像一下建立一個助理，能夠在線研究主題，總結發現，並自動將其直接保存到你的 Notion。這就是 AI 代理所能實現的智能自動化。

But here's the real challenge: getting an AI to reliably act in the real world—interacting with APIs, scraping websites, updating databases. How do you bridge the gap between the AI's reasoning and the tools it needs to execute real tasks?

但真正的挑戰在於：讓 AI 能夠可靠地在現實世界中行動——與 API 互動、抓取網站、更新資料庫。你如何彌合 AI 推理與執行實際任務所需工具之間的鴻溝？

In this guide, we'll discuss three solid ways to build AI agents:

在本指南中，我們將討論三種構建 AI 代理的可靠方法：

- From scratch (if you're feeling hardcore),

  從零開始（如果你想挑戰自我），

- With frameworks like LangChain and CrewAI (if you want flexibility without reinventing the wheel),

  使用像 LangChain 和 CrewAI 這樣的框架（如果你想靈活運用而不必重新發明輪子），

- Or with n8n (if you like visual workflows and want to build fast, production-ready AI agents).

  或使用 n8n（如果你喜歡視覺化工作流程並希望快速構建可投入生產的 AI 代理）。

We'll keep it hands-on, and by the end, you'll have a working AI agent that actually does stuff — not just thinks about it.

我們將保持實作導向，最後你將擁有一個真正能執行任務的 AI 代理，而不僅僅是思考它。

Understanding the basics of building AI Agents

理解構建 AI 代理的基礎知識

Before diving into building, let's break down how an AI agent works.

在開始構建之前，讓我們先拆解 AI 代理的運作原理。

At its core, an AI agent acts as a system that acts on behalf of a user (or program) to achieve a specific result by perceiving its environment, making decisions, and taking actions. While they can range from simple chatbots to complex autonomous systems, most AI agents share a few fundamental components:

本質上，AI 代理是一個代表使用者（或程式）行動的系統，透過感知環境、做出決策並採取行動來達成特定目標。雖然它們可以從簡單的聊天機器人到複雜的自主系統不等，但大多數 AI 代理都共享一些基本組成部分：

Perception 感知

It's the ability to gather information from its environment. This environment could be a chat interface, a database, a webpage, or even physical sensors. Inputs might include:

它是從環境中收集資訊的能力。這個環境可以是聊天介面、資料庫、網頁，甚至是實體感測器。輸入可能包括：

- Text commands from a user (i.e. a message or prompt).
  使用者的文字指令（例如訊息或提示）。
- Events triggered by other systems, such as webhooks or messages.
  由其他系統觸發的事件，例如 webhook 或訊息。
- Information retrieved from websites or APIs.
  從網站或 API 獲取的信息。

- Content from documents or databases.

  來自文件或資料庫的內容。

Decision-making 決策制定

This is the agent's "brain." Based on its perception (the information gathered) and its programmed goals, the agent decides what to do next. This core logic can involve:

這是代理的「大腦」。根據其感知（收集到的信息）和其程式設定的目標，代理決定下一步該做什麼。這個核心邏輯可以包括：

- Large Language Models (LLMs)**:** Modern agents often leverage LLMs (like GPT, Gemini, Claude etc.) as their primary reasoning engine to understand requests, formulate plans, and generate responses.

  大型語言模型（LLMs）：現代代理通常利用 LLMs（如 GPT、Gemini、Claude 等）作為其主要的推理引擎，以理解請求、制定計劃並生成回應。

- Rule-Based Systems**:** Simple instructions like "if the customer asks for a refund, execute the refund workflow."

  規則基礎系統：簡單的指令，例如「如果客戶要求退款，則執行退款工作流程」。

- Machine Learning Models**:** Algorithms trained to predict outcomes or classify information to guide decisions.

  機器學習模型：訓練用於預測結果或分類資訊以指導決策的演算法。

Planning**:** Breaking down a complex goal (e.g., "plan a trip to Rome") into smaller, manageable steps (search flights, find hotels, check visa requirements).

規劃：將複雜目標（例如「規劃一次羅馬之旅」）拆解為較小且可管理的步驟（搜尋航班、尋找飯店、檢查簽證要求）。

Action 行動

Once a decision is made, the agent needs to act upon it. This involves interacting with its environment to execute the chosen steps. Actions can be diverse, such as:

一旦做出決定，代理人需要根據該決定採取行動。這涉及與其環境互動以執行所選擇的步驟。行動可以多種多樣，例如：

- Sending a message back to the user.

  向使用者回傳訊息。
- Calling an API (like searching the web or posting to to a Discord channel).

  呼叫 API（例如搜尋網路或發佈到 Discord 頻道）。
- Running a workflow (like an n8n workflow!).

  執行工作流程（例如 n8n 工作流程！）。
- Updating information in a database.

  更新資料庫中的資訊。
- Controlling a physical device. Actions are how the agent influences its environment to move closer to its goal. The ability to use various AI agent tools (like APIs or workflows) is central to an agent's effectiveness.

  控制實體裝置。行動是代理如何影響其環境以更接近目標的方式。能夠使用各種 AI 代理工具（如 API 或工作流程）是代理有效性的核心。

When an agent uses an LLM for decision-making, the LLM needs a structured way to understand which actions it can take and how to execute them. This is often achieved through defining **Tools** or enabling **Function Calling**. These mechanisms allow the LLM to signal its intent to use a specific capability (like calling an external API or running an n8n workflow) and provide the necessary parameters.

當代理使用 LLM 進行決策時，LLM 需要一種結構化的方式來理解它可以採取哪些行動以及如何執行這些行動。這通常是通過定義工具或啟用函數調用來實現的。這些機制允許 LLM 表達其使用特定功能（如調用外部 API 或運行 n8n 工作流程）的意圖，並提供必要的參數。

Memory 記憶

Agents often need to remember past interactions or learned information to provide context for future decisions. Memory allows an agent to:

代理人經常需要記住過去的互動或學習到的信息，以便為未來的決策提供上下文。記憶使代理人能夠：

- Recall previous parts of a conversation to maintain context.

  回憶對話的先前部分以維持上下文。

- Store user preferences (e.g., "always use metric units").

  儲存使用者偏好（例如，「始終使用公制單位」）。

- Access external knowledge bases (like documents or databases) to answer questions accurately (often using techniques like Retrieval-Augmented Generation or RAG).

  存取外部知識庫（如文件或資料庫）以準確回答問題（通常使用檢索增強生成技術，或稱 RAG）。

- Learn from past experiences to improve future performance.

  從過去的經驗中學習，以提升未來的表現。

These components work together in a continuous loop: the agent perceives its environment, decides on an action based on its goals and memory, and then performs that action, potentially changing the environment and starting the cycle again.

這些組件在一個持續的循環中協同運作：代理感知其環境，根據其目標和記憶決定行動，然後執行該行動，可能改變環境並重新開始循環。

Understanding these basic building blocks is the first step. Next, let's look at the different ways you can actually build AI agents.

理解這些基本構建模塊是第一步。接下來，讓我們來看看實際構建 AI 代理的不同方法。

💡 Interested in seeing AI agents in action? This article provides 15 real-world examples of how AI agents, particularly those built with n8n, are automating tasks like data analysis, customer support, and more!

有興趣看看 AI 代理的實際應用嗎？本文提供了 15 個真實案例，展示了特別是使用 n8n 建立的 AI 代理如何自動化執行資料分析、客戶支援等任務！

How to create AI agents: 3 practical approaches
如何創建 AI 代理：三種實用方法

So, how do we actually go about building an AI agent? There are several ways to approach this, each with its own set of trade-offs in terms of flexibility, complexity, and development speed.

那麼，我們究竟該如何構建一個 AI 代理呢？有多種方法可以採用，每種方法在靈活性、複雜性和開發速度方面各有利弊。

Let's look at three common methods:

讓我們來看看三種常見的方法：

Building AI agents from scratch
從零開始構建人工智慧代理

Learning how to build an AI agent from scratch involves coding all the components using programming languages like Python and potentially leveraging specific AI/ML

libraries.

學習如何從零開始構建人工智慧代理涉及使用像 Python 這樣的程式語言編寫所有組件，並可能利用特定的人工智慧/機器學習庫。

This approach offers maximum flexibility and control over every aspect of the agent's behavior.

這種方法在代理行為的每個方面提供了最大的靈活性和控制力。

However, it requires significant technical expertise in areas like software engineering, API integration, and potentially machine learning. It also demands considerable development time and effort to build, test, and maintain the entire system.

然而，這需要在軟體工程、API 整合以及可能的機器學習等領域具備顯著的技術專長。它還需要大量的開發時間和精力來構建、測試和維護整個系統。

This path often answers the question 'how much does it cost to build an AI agent?' with 'significantly', due to the required development time and expertise. It is often chosen for highly specialized or research-oriented projects where existing tools don't meet specific requirements.

由於所需的開發時間和專業知識，這條路徑常常對「建立一個 AI 代理需要多少成本？」這個問題的回答是「相當高」。它通常被用於高度專業化或研究導向的專案，當現有工具無法滿足特定需求時會被選擇。

Using existing frameworks for building AI agents 使用現有框架構建人工智慧代理人

Several frameworks (like LangChain, LlamaIndex, Semantic Kernel, or Autogen) provide pre-built components and abstractions designed for creating AI agents. These frameworks offer building blocks for managing prompts, connecting to LLMs, handling memory, defining tools (actions), and orchestrating agent steps. They significantly accelerate development compared to building from scratch by handling much of the

underlying complexity.

幾個框架（如 LangChain、LlamaIndex、Semantic Kernel 或 Autogen）提供了為創建 AI agents 設計的預建組件和抽象層。這些框架提供了管理提示、連接到 LLMs、處理記憶、定義工具（行動）以及協調 agent 步驟的構建模塊。與從零開始構建相比，它們通過處理大部分底層複雜性，大大加快了開發速度。

However, they still require coding proficiency and a good understanding of the chosen framework's architecture and concepts.

然而，它們仍然需要具備編碼能力以及對所選框架架構和概念的良好理解。

This approach strikes a balance between flexibility and development speed, suitable for teams wanting structured development with some customization.

這種方法在靈活性和開發速度之間取得平衡，適合希望在結構化開發中進行一定程度自訂的團隊。

> 💡 We've rounded up 9 popular AI agent frameworks—from drag-and-drop simplicity to fully code-driven setups. Each one offers a different level of control, complexity, and customization.
>
> 我們整理了 9 個熱門的 AI 代理框架，從拖放式簡單操作到完全程式碼驅動的設置。每一個都提供不同程度的控制、複雜性和自訂化。

Using workflow automation tools
使用工作流程自動化工具

Platforms like n8n provide a visual, node-based environment for building agents. You connect services like LLMs, APIs, and databases as nodes, defining the agent's logic

and actions by arranging and configuring these nodes in a workflow.

像 n8n 這樣的平台提供了一個視覺化的節點式環境來構建代理。您可以將 LLMs、API 和資料庫等服務作為節點連接，通過排列和配置這些節點來定義代理的邏輯和行動流程。

This approach significantly lowers the barrier to entry and speeds up development and prototyping, shifting the focus from complex coding to workflow design and tool integration.

這種方法大幅降低了入門門檻，加快了開發和原型設計的速度，將重點從複雜的程式編碼轉移到工作流程設計和工具整合上。

It's particularly well-suited for automating tasks, rapidly building agent prototypes, and integrating AI capabilities into broader business processes.

它特別適合用於自動化任務、快速構建代理原型，以及將人工智慧功能整合到更廣泛的業務流程中。

How to build an AI agent with n8n: Step-by-step tutorial

如何使用 n8n 建立人工智慧代理：逐步教學指南

n8n stands out as a choice for building AI agents because it uniquely balances implementation flexibility with speed of delivery. Although primarily a workflow automation tool, it allows for the creation of agents that can call multiple pre-built or custom tools, integrate RAG capabilities for knowledge retrieval and hook up to various

chat interfaces via its flexible API and SDK options.

n8n 脫穎而出成為構建 AI 代理的選擇，因為它獨特地平衡了實現的靈活性與交付速度。儘管它主要是一個工作流程自動化工具，但它允許創建能夠調用多個預建或自定義工具的代理，整合用於知識檢索的 RAG 功能，並通過其靈活的 API 和 SDK 選項連接到各種聊天介面。

We’re going to build a practical research agent that scrapes the web and saves the summary for us—automatically!

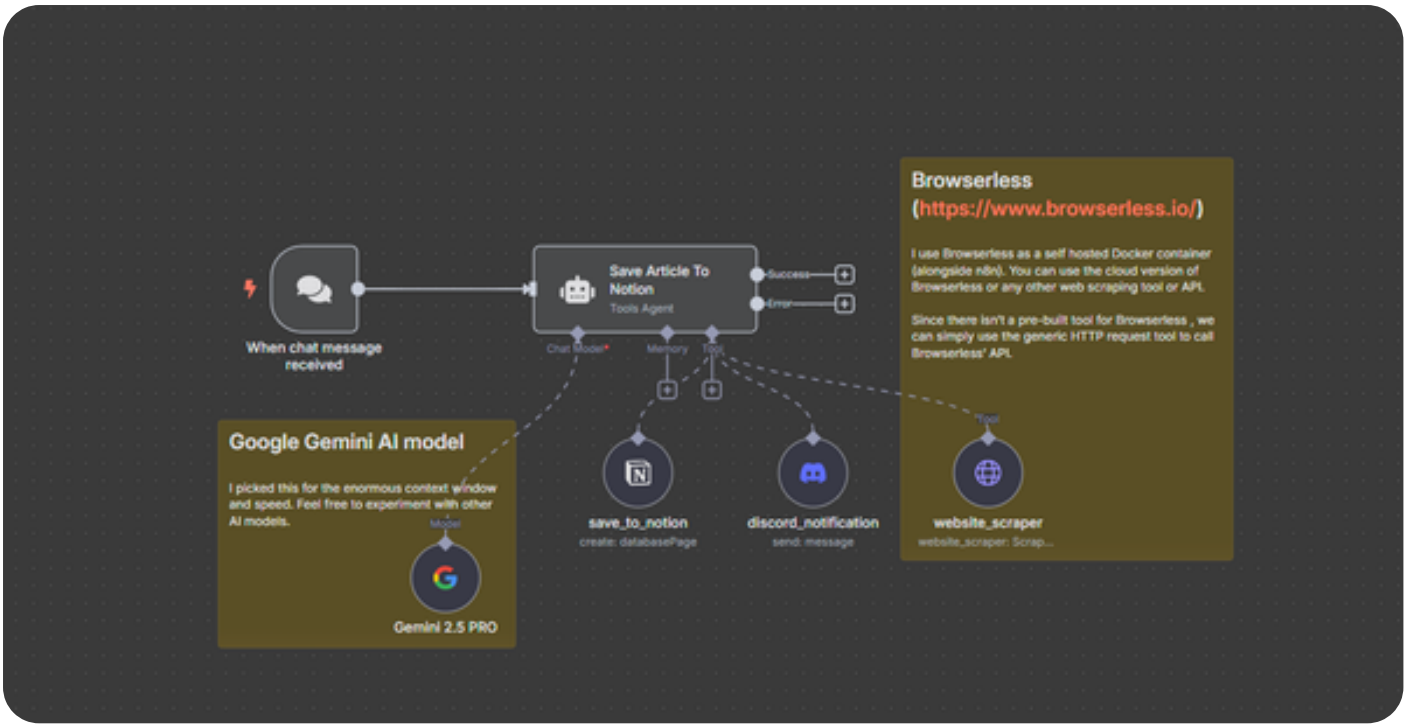
我們將構建一個實用的研究代理，自動從網路抓取資料並為我們保存摘要！

Since AI agents' main functionality is the use of tools such as HTTP requests and the Notion tool – this example leverages advanced LLMs like the recently released Gemini 2.5 Pro, whose reliable tool calling and enormous context window make such tasks feasible!

由於 AI 代理的主要功能是使用如 HTTP 請求和 Notion 工具等工具——此範例利用了先進的 LLMs，如最近發布的 Gemini 2.5 Pro，其可靠的工具調用和巨大的上下文視窗使這類任務成為可能！

Here is an image of what we will build using n8n:

這是我們將使用 n8n 建立的圖像：



n8n research agent workflow

n8n 研究代理工作流程

AI Agent: Scrape, Summarize & Save Articles to Notion (Gemini, Browserless)

AI 代理：抓取、摘要並保存文章至 Notion（Gemini，無瀏覽器模式）

by mihailtd 由 mihailtd 提供

Use this workflow

使用此工作流程

Let's break this down! 讓我們來拆解看看！

Prerequisites 先決條件

Before we start building the workflow, ensure you have the following set up:

在開始建立工作流程之前，請確保您已完成以下設定：

- **n8n instance:** You need n8n running. This can be a self-hosted instance (e.g., using Docker) or an account on n8n Cloud.

  n8n 實例：您需要運行 n8n。這可以是自我托管的實例（例如，使用 Docker）或 n8n Cloud 上的帳戶。

- **Browserless:** Access to a Browserless instance is required for web scraping. You can use their cloud service or self-host your own instance (e.g., using Docker).

Browserless：進行網頁爬蟲需要存取 Browserless 實例。您可以使用他們的雲端服務或自行架設實例（例如，使用 Docker）。

- **Google AI API Key:** Obtain an API key from Google AI Studio to use the Gemini model.

  Google AI API 金鑰：從 Google AI Studio 獲取 API 金鑰以使用 Gemini 模型。

- **Discord**: Configure a Discord webhook or bot account to send notification when research is done.

  Discord：配置 Discord webhook 或機器人帳號，以在研究完成時發送通知。

Step 1: Set up the trigger

第一步：設定觸發器

Every n8n workflow starts with a trigger node. This node initiates the workflow when a specific event occurs. For our AI Research Agent, we want it to activate when we send it a message containing a URL, typically via a chat interface.

每個 n8n 工作流程都從觸發節點開始。當特定事件發生時，此節點會啟動工作流程。對於我們的 AI 研究代理，我們希望它在收到包含 URL 的訊息時啟動，通常是透過聊天介面。

In the n8n canvas, click the '+' button to add your first node. Choose a trigger relevant to how you want to interact with your agent. In this case we can use the Chat trigger. Other common triggers for such a use case would be the Webhook trigger, which creates a unique URL to which you can send HTTP requests from a custom application or another service, or the Slack trigger which listens for messages or commands in Slack.

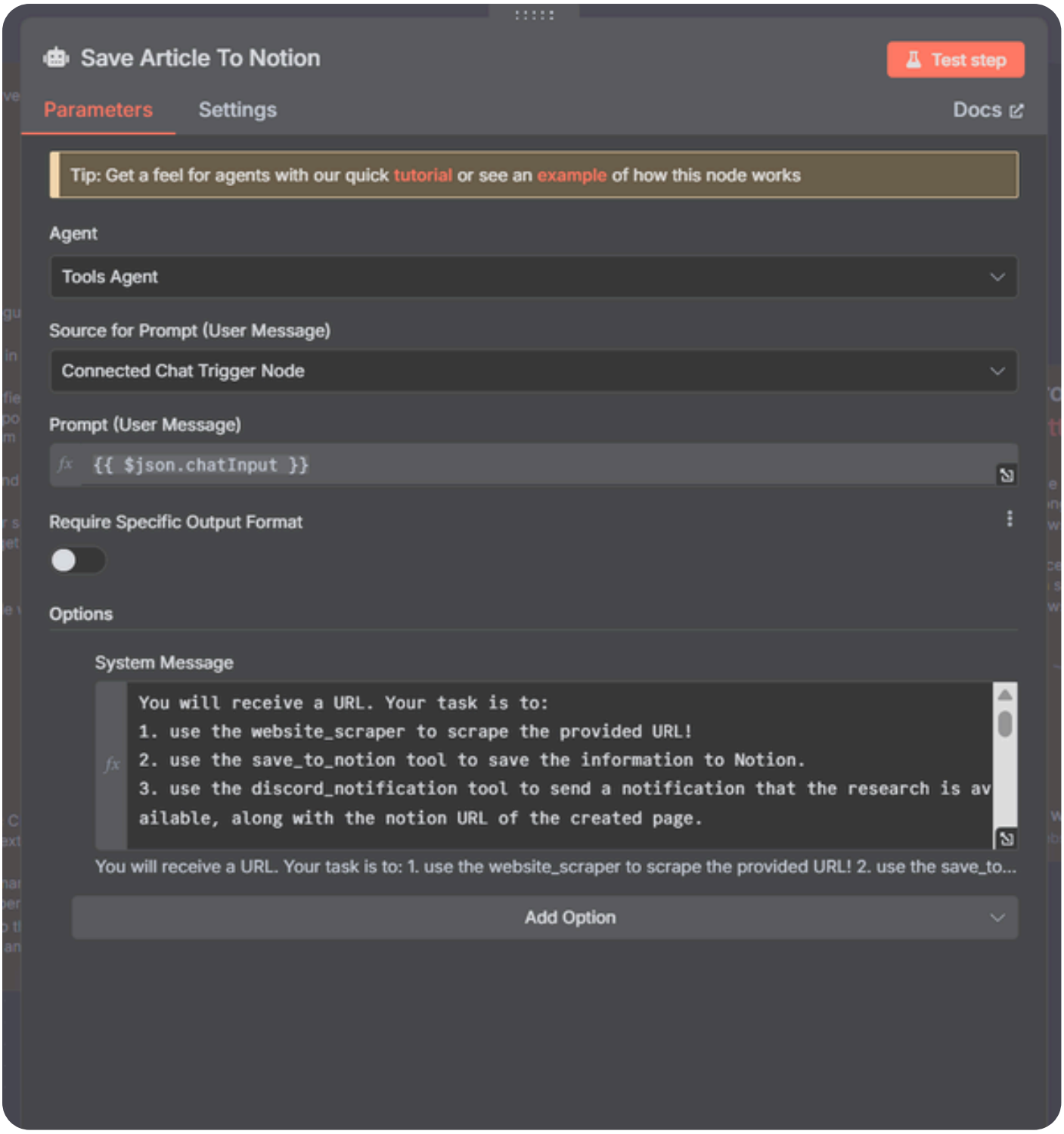
在 n8n 畫布中，點擊「+」按鈕以新增您的第一個節點。選擇一個與您想要與代理互動方式相關的觸發器。在此案例中，我們可以使用聊天觸發器。其他常見的觸發器包括 Webhook 觸發器，它會建立一個唯一的 URL，您可以從自訂應用程式或其他服務向該 URL 發送 HTTP 請求，或是 Slack 觸發器，該觸發器會監聽 Slack 中的訊息或指令。

Step 2: Configure the Agent's core

步驟二：配置代理的核心部分

The heart of our workflow is the AI Agent node. This node acts as the central orchestrator, connecting the trigger, the Large Language Model (LLM), and the tools the agent can use.

我們工作流程的核心是 AI Agent 節點。此節點作為中央協調者，連接觸發器、LLM 以及代理可使用的工具。



AI Agent node configuration

AI 代理節點配置

First, add an AI Agent node to the canvas and connect the output of your trigger node to the input of the AI Agent node.

首先，將 AI 代理節點添加到畫布上，並將觸發節點的輸出連接到 AI 代理節點的輸入。

In the AI Agent node settings (see image), ensure the **Agent** dropdown is set to *Tools Agent*. This type is designed for agents that need to utilize specific tools to accomplish tasks. Set the **Source for Prompt (User Message)** to *Connected Chat Trigger Node*. This tells the agent to use the input from your trigger (e.g., the chat message containing the URL) as the user's request. The specific input field might vary depending on your trigger node's output.

在 AI Agent 節點設定中（見圖片），確保 Agent 下拉選單設為 Tools Agent。此類型專為需要使用特定工具來完成任務的代理設計。將提示來源（使用者訊息）設為 Connected Chat Trigger Node。這告訴代理使用觸發器的輸入（例如包含 URL 的聊天訊息）作為使用者的請求。具體的輸入欄位可能會根據您的觸發節點輸出而有所不同。

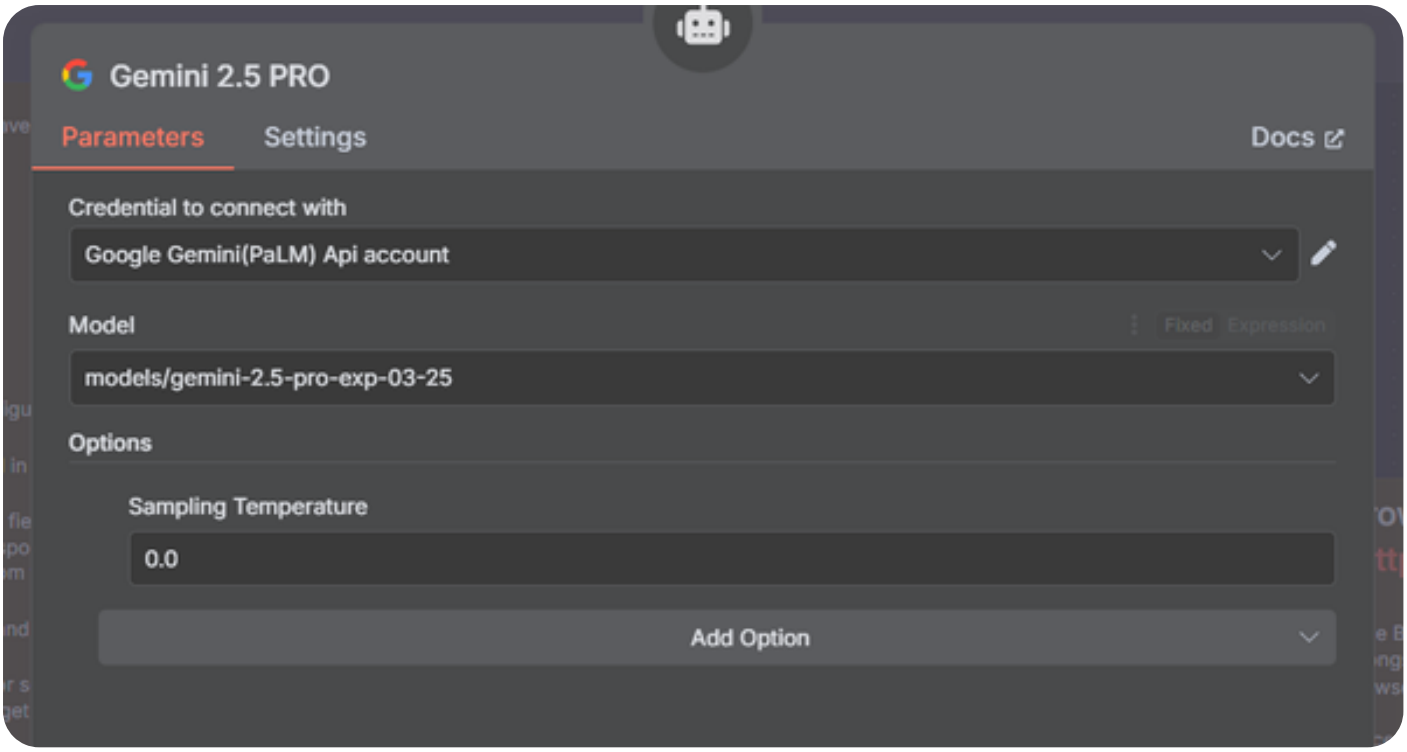
Step 3: Define the Agent's goal and instructions
第三步：定義代理的目標和指示

This is where you tell the AI what you want it to do and how it should use its tools. Clear instructions are crucial for reliable agent performance.

這裡是告訴 AI 你希望它做什麼以及如何使用其工具的地方。清晰的指示對於代理的可靠表現至關重要。

Add a Google Gemini Chat Model node (or your preferred LLM node like OpenAI Chat Model, Anthropic Chat Model, etc.) and configure it with your credentials (your Google AI API Key).

新增一個 Google Gemini Chat Model 節點（或您偏好的 LLM 節點，如 OpenAI Chat Model、Anthropic Chat Model 等），並使用您的憑證（您的 Google AI API 金鑰）進行設定。

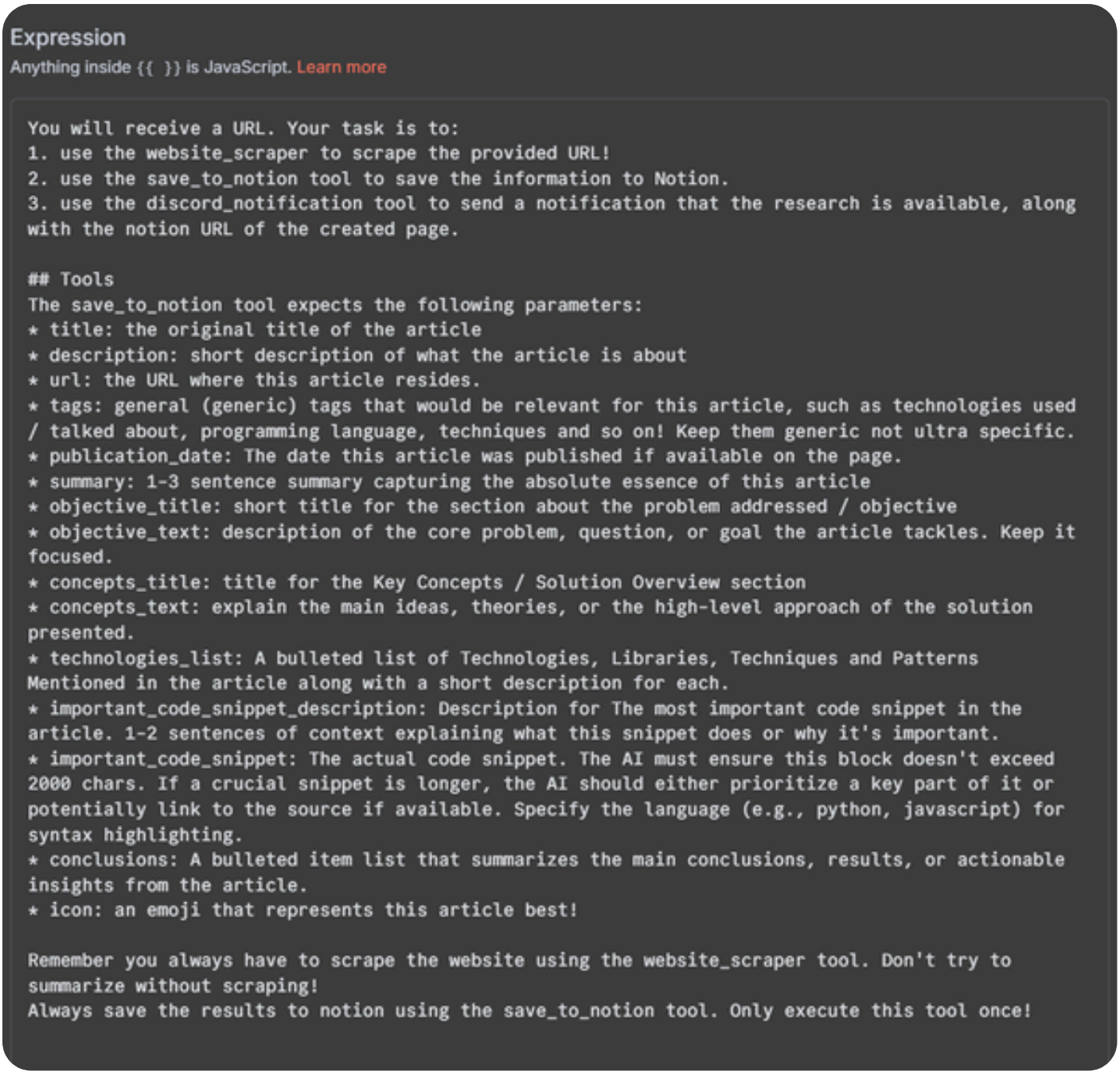


Google Gemini LLM configuration

Google Gemini LLM 配置

Select the desired Model (e.g., gemini-2.5-pro) and connect this LLM node to the Chat Model input of the AI Agent node.

選擇所需的模型（例如，gemini-2.5-pro），並將此 LLM 節點連接到 AI 代理節點的聊天模型輸入端。



Configure the system prompt

配置系統提示

In the AI Agent node parameters, there is a field called **System Message** within the **Options** section. This is where you provide the core instructions for the agent. Here you can define the agent's instructions. For best results, the system message should:

在 AI Agent 節點參數中，有一個名為「系統訊息」的欄位，位於選項區段。這裡是您提供代理核心指令的地方。在此您可以定義代理的指令。為達最佳效果，系統訊息應該：

1. Clearly state the agent's task.

   清楚說明代理人的任務。

2. Explicitly instruct the agent on when and how to use each tool.

   明確指示代理人何時以及如何使用每個工具。

3. Add any important constraints such as "Remember you always have to scrape the website using the website_scraper tool.", "Don't try to summarize without scraping!" etc.

   添加任何重要的限制條件，例如「請記住你必須始終使用 website_scraper 工具來抓取網站資料。」「未抓取資料前，請勿嘗試摘要！」等。

💡 Read more about using AI Tools in n8n.

閱讀更多關於在 n8n 中使用 AI 工具的資訊。

Step 4: Add the web scraping tool

第四步：新增網頁爬蟲工具

Now we configure the actual tools the agent can use. First, the web scraper uses Browserless. Since there isn't a dedicated Browserless node, we use the versatile HTTP Request Tool node.

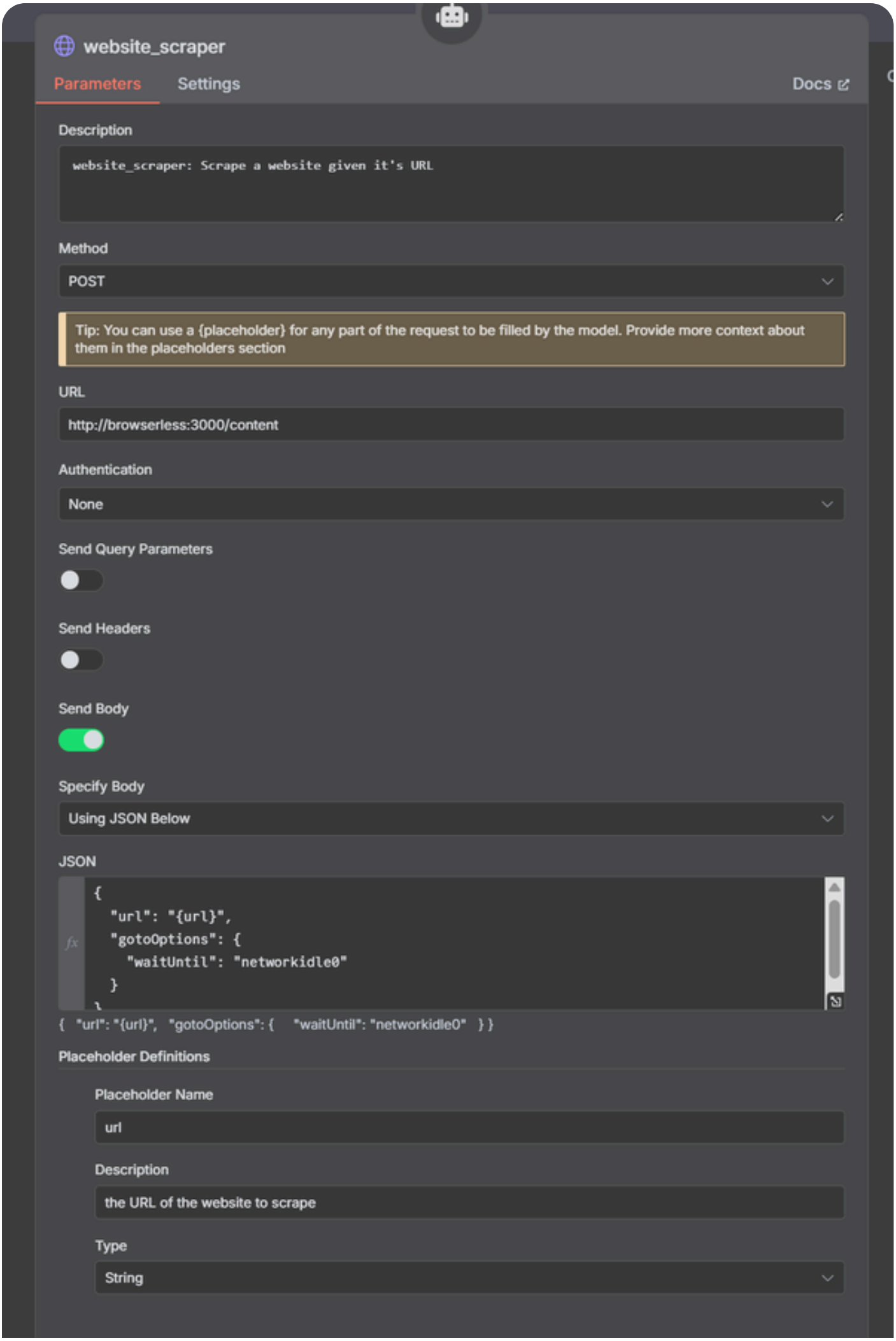
現在我們配置代理人可以使用的實際工具。首先，網頁爬蟲使用 Browserless。由於沒有專用的 Browserless 節點，我們使用多功能的 HTTP 請求工具節點。

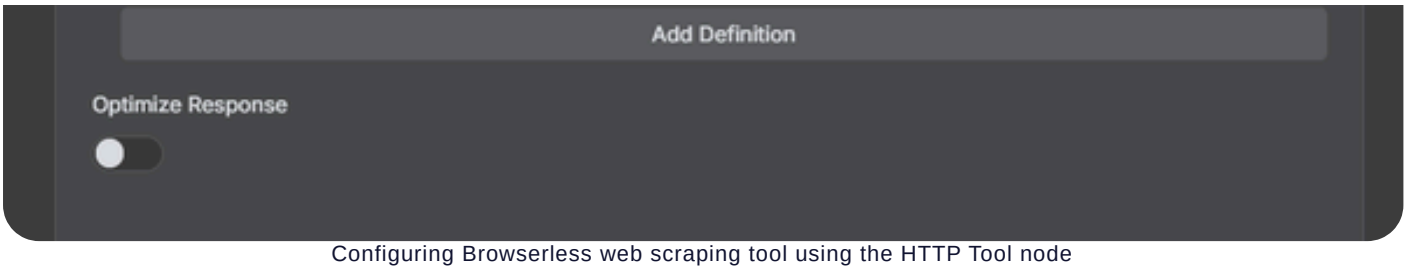
Add an HTTP Request Tool node to the canvas and rename it to *website_scraper* (or similar). This name must match the tool name used in the AI Agent's system message.

在畫布上新增一個 HTTP 請求工具節點，並將其重新命名為 website_scraper（或類似名稱）。此名稱必須與 AI Agent 系統訊息中使用的工具名稱相符。

💡 Choose a clear name and description for the tool; this significantly improves the chances the LLM will use it correctly.

為該工具選擇一個清晰的名稱和描述；這將大幅提升 LLM 正確使用該工具的機率。


website_scraper
Parameters
Settings
Docs
Description
website_scraper: Scrape a website given it's URL
Method
POST
Tip: You can use a {placeholder} for any part of the request to be filled by the model. Provide more context about them in the placeholders section
URL
http://browserless:3000/content
Authentication
None
Send Query Parameters
Send Headers
Send Body
Specify Body
Using JSON Below
JSON
{
"url": "{url}",
"gotoOptions": {
"waitUntil": "networkidle0"
}
}
{ "url": "{url}", "gotoOptions": { "waitUntil": "networkidle0" } }
Placeholder Definitions
Placeholder Name
url
Description
the URL of the website to scrape
Type
String



Configuring Browserless web scraping tool using the HTTP Tool node

使用 HTTP Tool 節點配置 Browserless 網頁爬蟲工具

Configure the node like in the image above:

如上圖所示配置節點：

- **Method**: *POST* 方法：POST
- **URL**: Enter your Browserless API endpoint for scraping content.

  URL：輸入您的 Browserless API 端點以抓取內容。
- **Authentication**: Configure if required by your Browserless setup.

  認證：如您的 Browserless 設定需要，請進行配置。
- **Body**: *Using JSON Below* 內容：使用以下 JSON
- **JSON**: Provide the JSON payload Browserless expects. Use a placeholder for the URL the agent will provide:

  JSON：提供 Browserless 所期望的 JSON 載荷。對代理將提供的 URL 使用佔位符：

```
{
  "url": "{url}",
  "gotoOptions": {
    "waitUntil": "networkidle0"
  }
}
```

- **Placeholder Definitions**: Define the placeholders used in the JSON body.

  佔位符定義：定義 JSON 主體中使用的佔位符。

  - Click **Add Definition**. 點擊新增定義。
  - **Placeholder Name**: *url* (must match the placeholder name in the JSON body and the parameter name expected by the AI Agent).

佔位符名稱：url（必須與 JSON 主體中的佔位符名稱及 AI 代理預期的參數名稱相符）。

- **Description**: Provide a clear description for the AI (e.g., "the URL of the website to scrape").

  描述：為人工智慧提供清晰的描述（例如，「要抓取的網站網址」）。

- **Type**: *String*. 類型：字串。

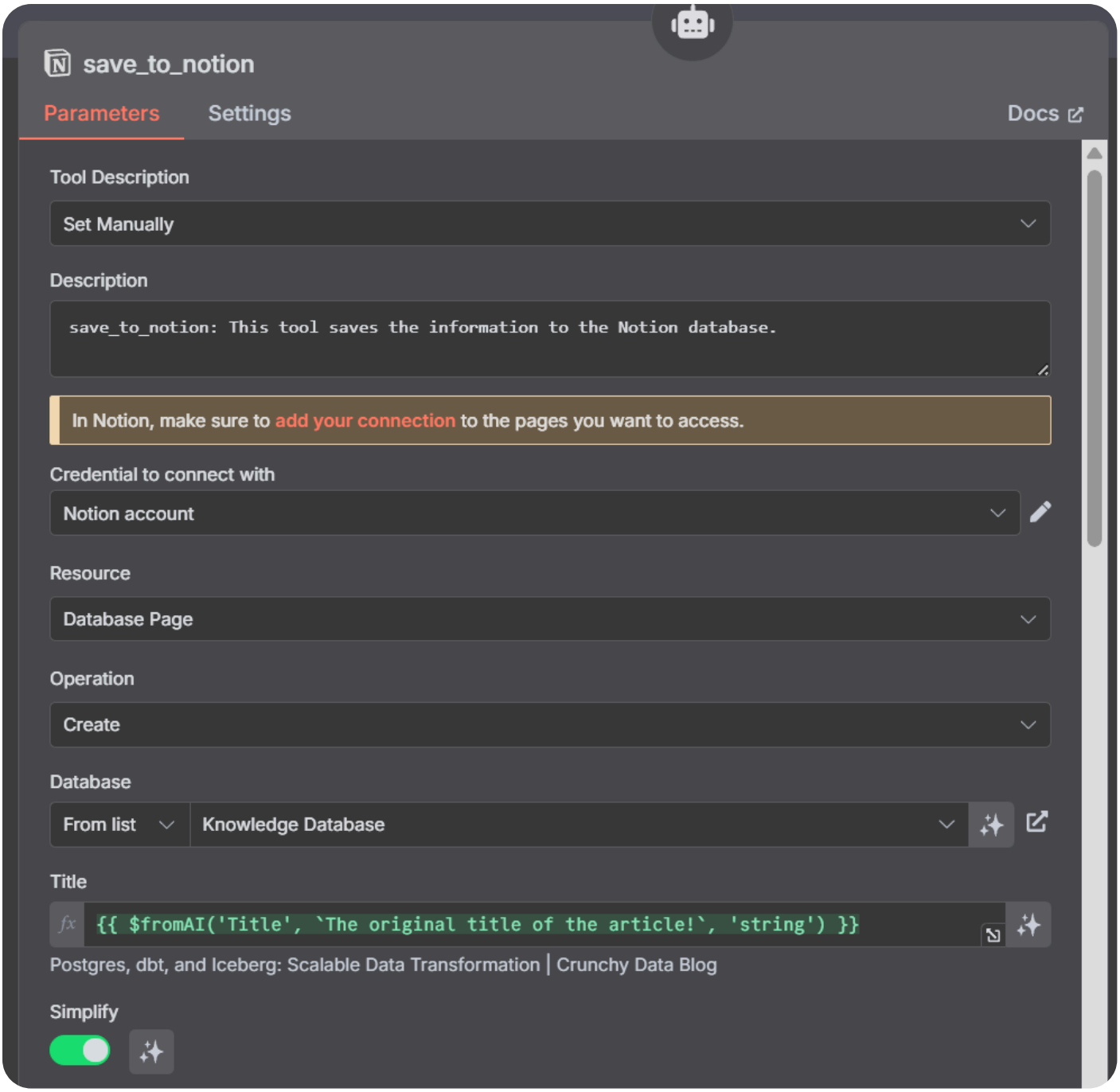
Finally, connect the HTTP Request Tool node (website_scraper) to the Tool input of the AI Agent node.

最後，將 HTTP 請求工具節點（website_scraper）連接到 AI 代理節點的工具輸入端。

Step 5: Define the save to Notion tool
第五步：定義儲存到 Notion 的工具

Next, configure the tool to save the scraped and summarized information to your Notion database.

接著，配置該工具以將擷取並摘要的資訊儲存到您的 Notion 資料庫中。



Configuring the Notion Tool node

配置 Notion 工具節點

Add a Notion Tool node to the canvas and rename it to *save_to_notion*, again, matching the tool name from the system message. Then, Set the **Tool Description** to *Manually* and provide a description (e.g., "save_to_notion: This tool saves the information to the Notion database.").

在畫布上新增一個 Notion Tool 節點，並將其重新命名為 save_to_notion，再次與系統訊息中的工具名稱相符。然後，將工具描述設為 Manually 並提供描述（例如：「save_to_notion：此工具將資訊保存到 Notion 資料庫中。」）。

For authenticating to Notion, Select your configured Notion API credentials. **Resource** should be set to *Database Page* and **Operation** to *Create.*

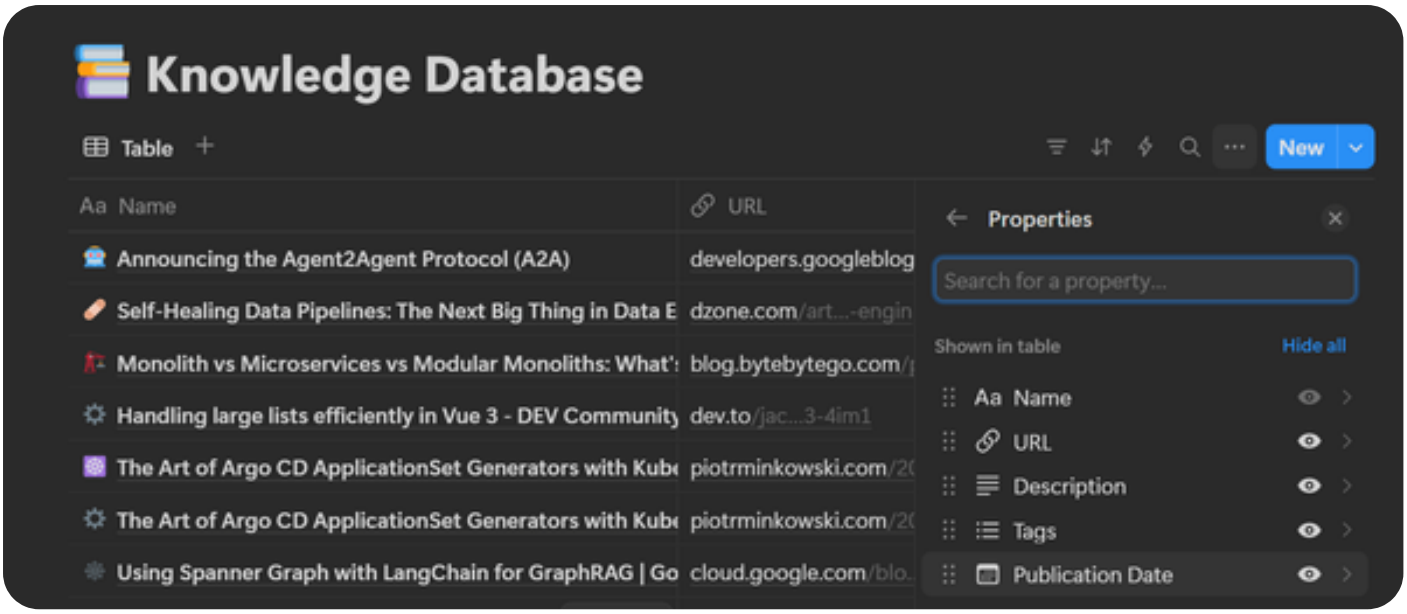
要驗證 Notion，請選擇您已配置的 Notion API 憑證。資源應設為資料庫頁面，操作應設為建立。

Select your target Notion database (e.g., "Knowledge Database"). Ensure the n8n integration has access to this database and that it contains the necessary properties (Name, URL, Description, Tags, etc., as defined in your setup). The **Properties** section is where you map the data generated by the AI Agent (based on the parameters defined in Step 3) to your Notion database fields. For each property (e.g., "Title", "Description", "URL", "Tags", "Publication Date", "Icon"), use the `{{ $fromAI('parameterName', 'Description', 'type') }}` expression. Replace `parameterName` with the exact parameter name you defined in the AI Agent's system message. Example for "Title": `{{ $fromAI('title', 'The original title of the article', 'string') }}`.

選擇您的目標 Notion 資料庫（例如，「知識資料庫」）。確保 n8n 整合有權限存取此資料庫，且該資料庫包含必要的屬性（名稱、URL、描述、標籤等，依您的設定定義）。屬性區段是您將 AI 代理生成的資料（根據步驟 3 中定義的參數）映射到 Notion 資料庫欄位的地方。對於每個屬性（例如「標題」、「描述」、「URL」、「標籤」、「發佈日期」、「圖示」），請使用 `{{ $fromAI('parameterName', 'Description', 'type') }}` 表達式。將 `parameterName` 替換為您在 AI 代理系統訊息中定義的精確參數名稱。「標題」的範例：`{{ $fromAI('title', 'The original title of the article', 'string') }}` 。

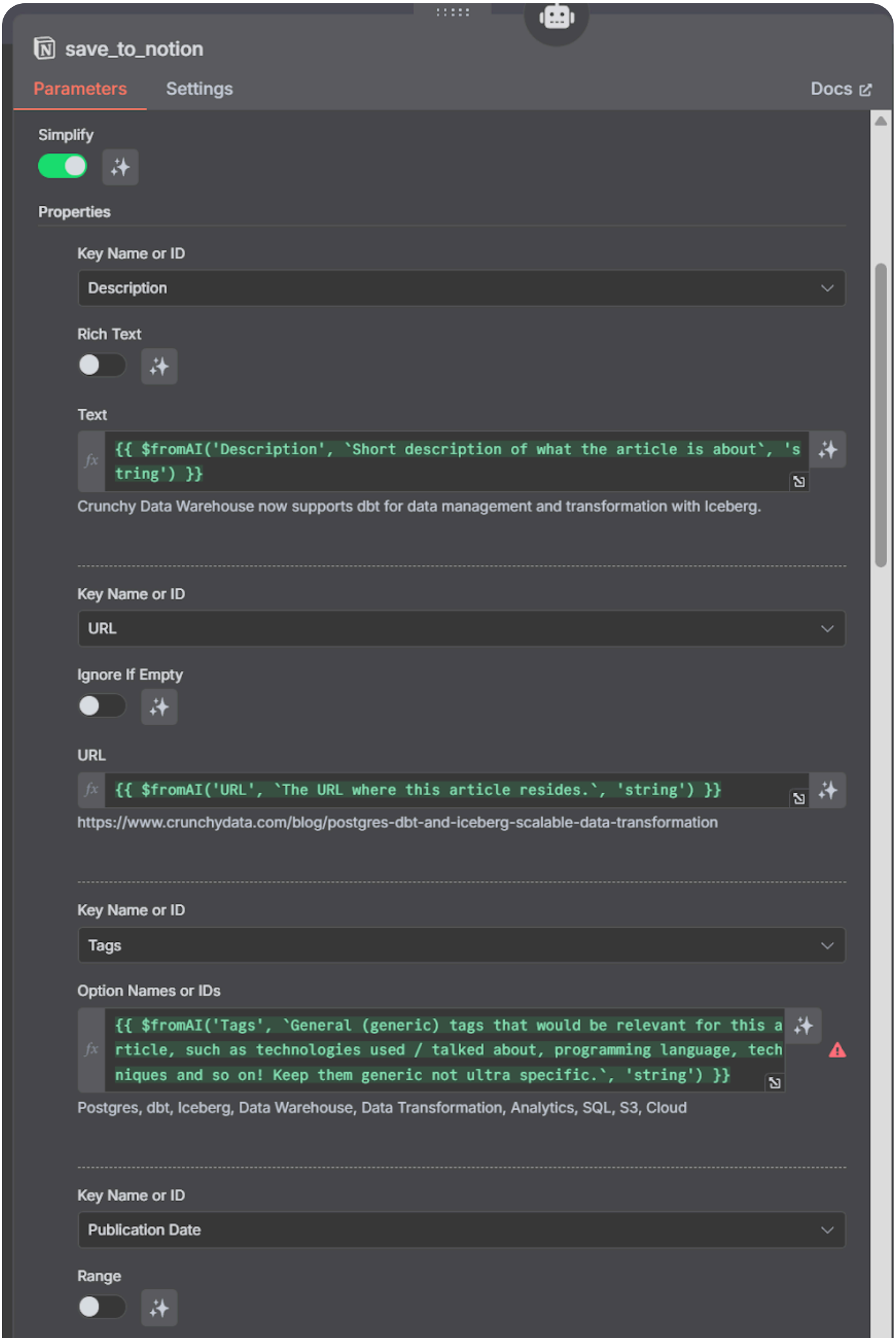
Here is how this specific Notion database is structured:

這是該特定 Notion 資料庫的結構方式：



Notion database structure

Notion 資料庫結構

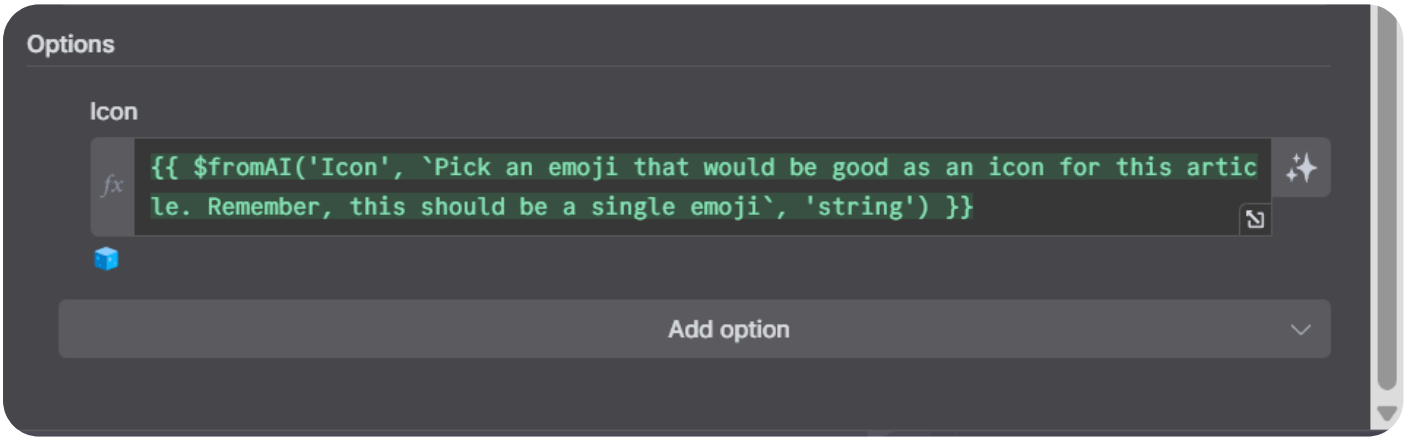

save_to_notion
Parameters
Settings
Docs
Simplify
Properties
Key Name or ID
Description
Rich Text
Text
{{ $fromAI('Description', `Short description of what the article is about`, 'string') }}
Crunchy Data Warehouse now supports dbt for data management and transformation with Iceberg.
Key Name or ID
URL
Ignore If Empty
URL
{{ $fromAI('URL', `The URL where this article resides.`, 'string') }}
https://www.crunchydata.com/blog/postgres-dbt-and-iceberg-scalable-data-transformation
Key Name or ID
Tags
Option Names or IDs
{{ $fromAI('Tags', `General (generic) tags that would be relevant for this article, such as technologies used / talked about, programming language, techniques and so on! Keep them generic not ultra specific.`, 'string') }}
Postgres, dbt, Iceberg, Data Warehouse, Data Transformation, Analytics, SQL, S3, Cloud
Key Name or ID
Publication Date
Range

Configure the AI generated fields for the Notion database page

為 Notion 資料庫頁面配置 AI 生成欄位

As a useful visual touch, let's prompt the AI agent to select a fitting emoji for each page:

作為一個有用的視覺點綴，讓我們提示 AI 代理為每個頁面選擇一個合適的表情符號：



Use AI to choose an emoji icon for the article

使用人工智慧為文章選擇表情符號圖示

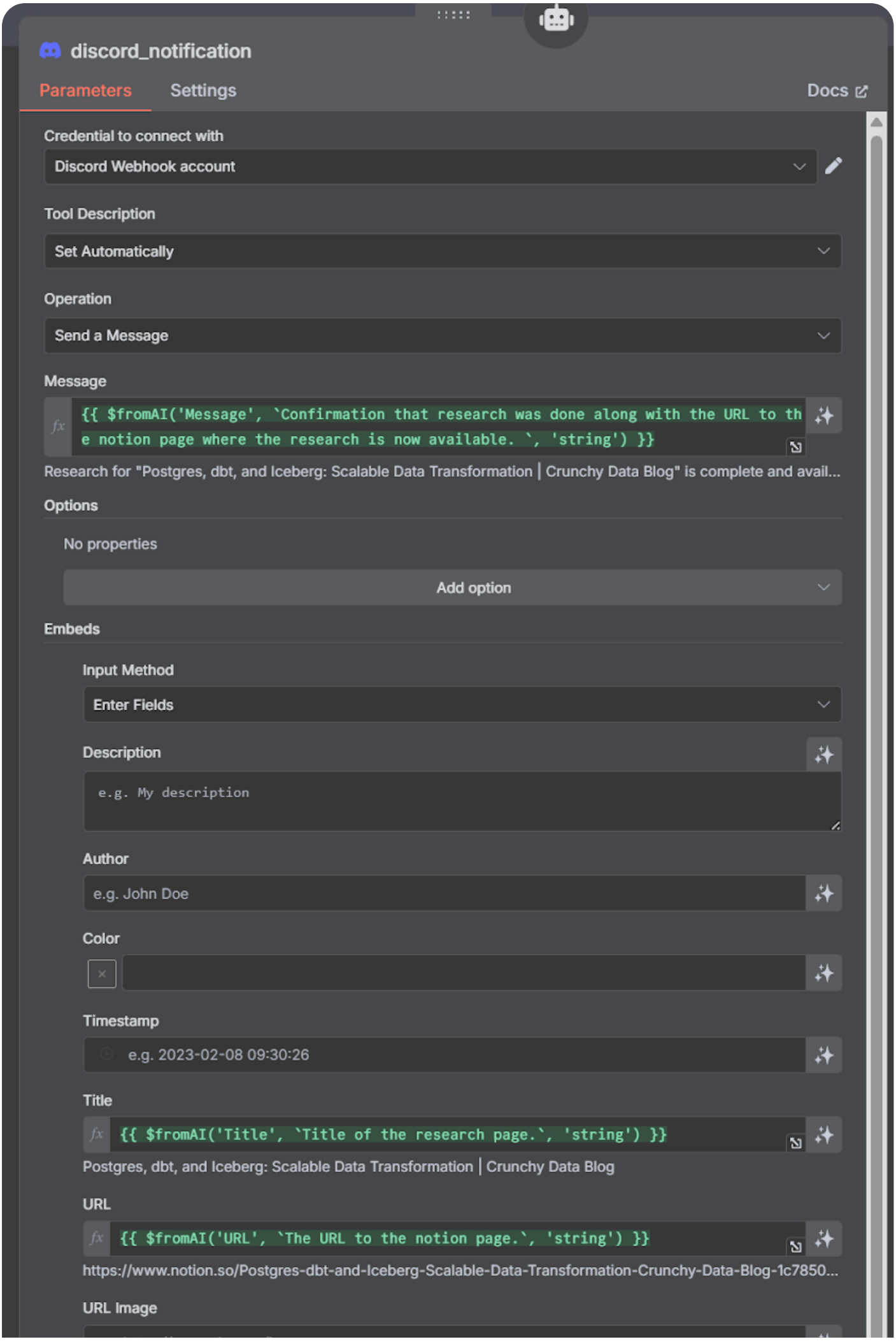
Step 6: Define the Discord notification tool

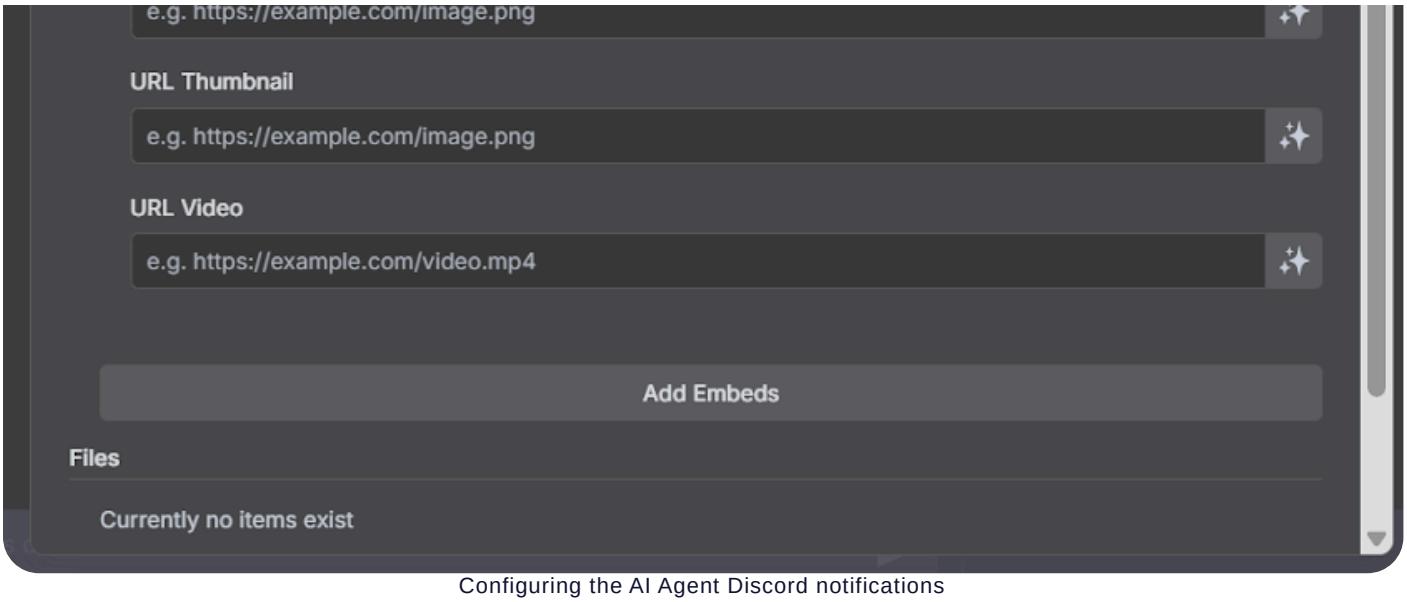
第六步：定義 Discord 通知工具

To ensure the AI agent can report its task completion, we'll equip it with a tool for sending Discord messages. This allows the agent itself to decide when and how to

notify you based on its instructions and the outcome of its tasks.

為確保 AI 代理能夠報告其任務完成情況，我們將為其配備一個發送 Discord 訊息的工具。這使得代理本身能根據其指令和任務結果決定何時以及如何通知您。


discord_notification
Parameters
Settings
Docs
Credential to connect with
Discord Webhook account
Tool Description
Set Automatically
Operation
Send a Message
Message
{{ $fromAI('Message', `Confirmation that research was done along with the URL to th
e notion page where the research is now available. `, 'string') }}
Research for "Postgres, dbt, and Iceberg: Scalable Data Transformation | Crunchy Data Blog" is complete and avail...
Options
No properties
Add option
Embeds
Input Method
Enter Fields
Description
e.g. My description
Author
e.g. John Doe
Color
Timestamp
e.g. 2023-02-08 09:30:26
Title
{{ $fromAI('Title', `Title of the research page.`, 'string') }}
Postgres, dbt, and Iceberg: Scalable Data Transformation | Crunchy Data Blog
URL
{{ $fromAI('URL', `The URL to the notion page.`, 'string') }}
https://www.notion.so/Postgres-dbt-and-Iceberg-Scalable-Data-Transformation-Crunchy-Data-Blog-1c7850...
URL Image



Configuring the AI Agent Discord notifications

配置 AI 代理的 Discord 通知

Add a Discord Tool node, name it "discord_notification" and select your Discord Webhook or Bot credentials. Select *Send a Message i*n the **Operation** dropdown. Here we will prompt the AI agent to craft the notification message, for example: `{{ $fromAI('Message', 'Confirmation that research was done along with the URL to the notion page where the research is now available.', 'string') }}`. And optionally in the **Embeds** field, we can embed a link for a richer notification by including the title and URL to the newly created Notion page.

新增一個 Discord 工具節點，命名為「discord_notification」，並選擇您的 Discord Webhook 或 Bot 憑證。在操作下拉選單中選擇「發送訊息」。在此，我們將提示 AI 代理撰寫通知訊息，例如： `{{ $fromAI('Message', 'Confirmation that research was done along with the URL to the notion page where the research is now available.', 'string') }}` 。另外，在「嵌入」欄位中，我們可以嵌入一個連結，以包含新建立的 Notion 頁面的標題和 URL，從而提供更豐富的通知內容。

Step 7: Test and refine your AI agent

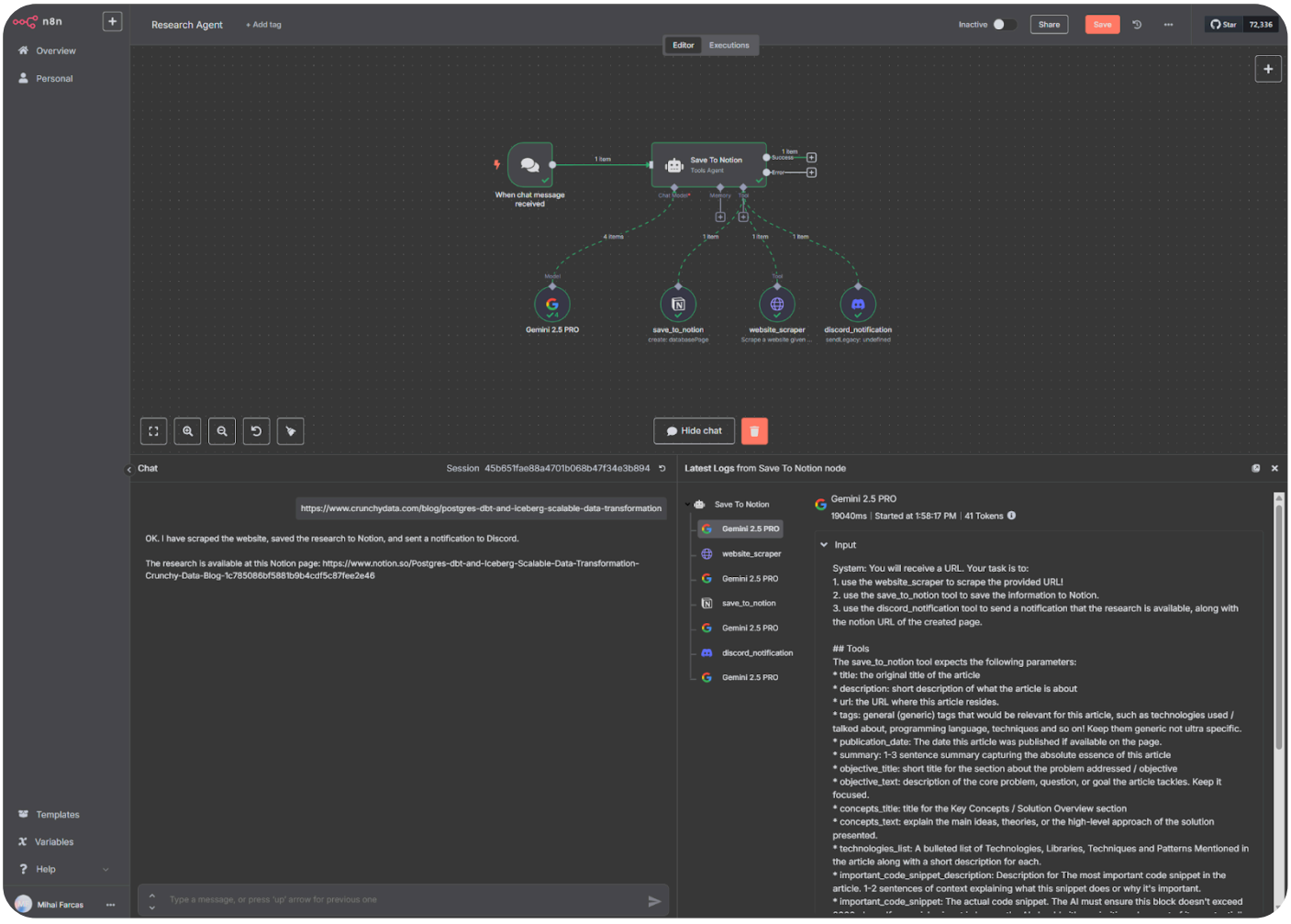
第七步：測試並優化您的 AI 代理

💡 Building AI agents is an iterative process. Testing thoroughly and refining your instructions are key to achieving reliable performance.

建立 AI 代理是一個反覆進行的過程。徹底測試並精煉您的指令是達成可靠效能的關鍵。

Make sure to save the workflow and send a chat message containing a URL you want researched. Observe the workflow execution in the n8n UI checking the input/output of each node, especially the AI Agent node, to see how it processes the request and which tools it decides to call.

請務必儲存工作流程，並發送包含您希望研究的 URL 的聊天訊息。在 n8n 使用者介面中觀察工作流程的執行，檢查每個節點的輸入/輸出，特別是 AI 代理節點，以了解它如何處理請求以及決定調用哪些工具。

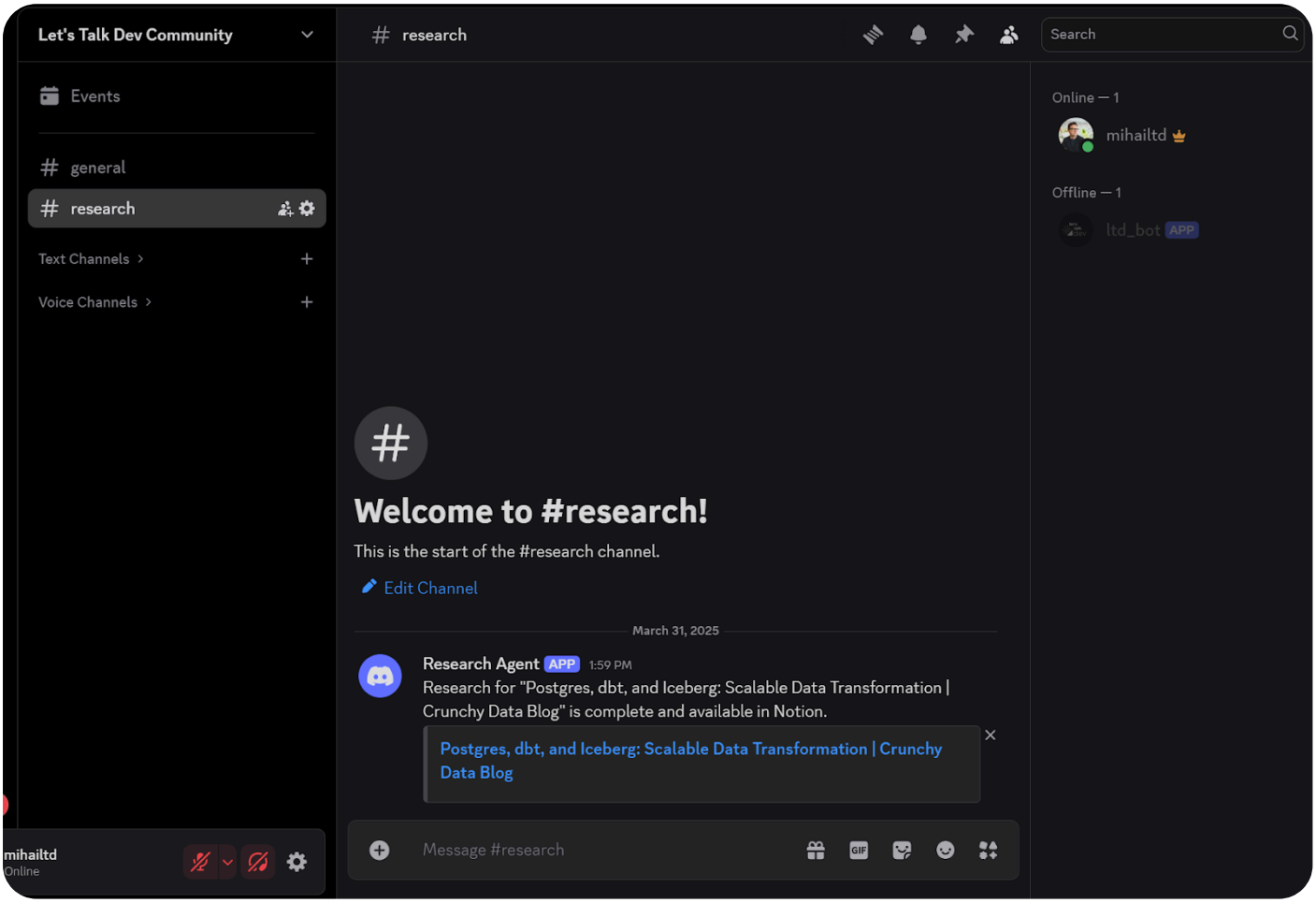


Testing the AI agent using the n8n UI

使用 n8n 使用者介面測試 AI 代理人

Verify that the website was scraped correctly, check that a new page was created in your Notion database with the expected content and summary and that you received a notification in Discord.

確認網站是否正確擷取，檢查您的 Notion 資料庫中是否已建立包含預期內容和摘要的新頁面，並且您是否在 Discord 收到通知。



Discord notification sent by the AI agent

由 AI 代理發送的 Discord 通知

If the agent doesn't perform as expected, check each node and tool output, looking for reasoning or any errors in tool calling. Modify the instructions in the **System Message**, make the instructions clearer, add constraints, or refine the parameter definitions. Then save, and test again until the AI Research Agent reliably performs the desired scrape,

summarize, save, and notify tasks.

如果代理人未如預期執行，請檢查每個節點和工具輸出，尋找推理過程或工具調用中的任何錯誤。修改系統訊息中的指令，使指令更清晰，添加限制條件，或精煉參數定義。然後保存並再次測試，直到 AI 研究代理人能可靠地執行所需的抓取、摘要、保存和通知任務。

Wrap up 結語

In this article, we looked at the core components of AI agents (like how they perceive, decide, act, and remember) and the main ways to architect them: coding from scratch, using frameworks, or using visual tools like n8n.

在本文中，我們探討了 AI 代理人的核心組成部分（如它們如何感知、決策、行動和記憶）以及架構它們的主要方式：從零編碼、使用框架，或使用像 n8n 這樣的視覺化工具。

Our research agent example showed how n8n makes it straightforward to create powerful agents. By connecting LLMs with different tools visually, you can build systems that act intelligently for you. Good tool integration is vital for agents, and n8n makes this easy.

我們的研究代理範例展示了 n8n 如何使建立強大代理變得簡單。透過視覺化地將 LLMs 與不同工具連接，您可以構建為您智能行動的系統。良好的工具整合對代理至關重要，而 n8n 使這一過程變得輕鬆。

Create your first AI agent
建立您的第一個人工智慧代理人

Use the power of n8n's flexibility to customize every step

利用 n8n 靈活性的力量，自訂每一個步驟

Try n8n now

現在就試用 n8n

AI agents are changing quickly, offering new ways to automate tasks and personalize experiences. We hope this guide clarified the basics and encourages you to try building your own agents with n8n.

AI 代理正在迅速變化，提供自動化任務和個性化體驗的新方法。我們希望本指南能釐清基礎知識，並鼓勵您嘗試使用 n8n 建立自己的代理。

What’s next? 接下來是什麼？

Now that you have a solid understanding of AI agents and how to start building them with n8n, you can experiment with different LLMs, try building agents for new use cases, and connect with the n8n community to share your creations and learn from others!

既然你已經對 AI 代理及如何使用 n8n 開始構建它們有了紮實的理解，你現在可以嘗試不同的 LLMs，嘗試為新的使用案例構建代理，並與 n8n 社群連結，分享你的創作並向他人學習！

You can also dive deeper and explore more resources from the community:

您也可以深入探索社群提供的更多資源：

- Check out this Reddit thread for insights on what tools to use for building AI agents.
  查看這個 Reddit 討論串，了解用於構建 AI 代理的工具見解。

- See how others are building with n8n in this YouTube video.

  在這個 YouTube 影片中看看其他人如何使用 n8n 進行構建。

- Read more on the n8n blog:

  閱讀更多關於 n8n 的部落格文章：

  - Learn about building AI agentic workflows.

    了解如何建立具代理性的 AI 工作流程。

  - Discover more on AI workflow automation.

    探索更多關於 AI 工作流程自動化的內容。

- Get inspired or find a starting point by browsing AI workflows shared by the n8n community.

  瀏覽 n8n 社群分享的 AI 工作流程，獲取靈感或找到起點。

Happy automating! 祝您自動化愉快！