**Notion knowledge base AI assistant.txt**

**Workflow: Notion Knowledge Base Assistant**

**Purpose:**  
This workflow generates a customized n8n workflow template that enables an AI agent to query a specific Notion database. By inputting a Notion database URL, the workflow produces a tailored JSON output representing a modified version of a base AI assistant template. This generated workflow can then be imported into n8n to provide a chatbot interface that interacts with the Notion database.

**Key Components**

1. Chat Trigger

* **Node:** *When chat message received*
  + **Type:** Chat Trigger (LangChain Chat Trigger)
  + **Description:**  
    This node listens for incoming chat messages via a webhook. When a user sends a message (typically containing a Notion database URL), the workflow is activated.
  + **Configuration:**
    - **Webhook URL:** Configured for public access.
    - **Initial Message:** Greets the user and prompts them to provide a Notion database URL.

2. Get Database Details

* **Node:** *Get database details*
  + **Type:** Notion (n8n Notion Node)
  + **Description:**  
    This node retrieves details of the specified Notion database. It uses the provided database URL (extracted from the user’s chat input) to fetch metadata about the database, such as its ID and properties.
  + **Configuration:**
    - **Database ID:** Extracted from the chat input using a regex matching URL.

3. Format Schema

* **Node:** *Format schema*
  + **Type:** Set (n8n Set Node)
  + **Description:**  
    This node formats and standardizes the input data required for the AI agent. It collects key information like the session ID, action, chat input, Notion database ID, and database name.
  + **Output:**  
    A standardized JSON object used as input for further processing.

4. AI Agent

* **Node:** *AI Agent*
  + **Type:** LangChain Agent Node
  + **Description:**  
    The AI Agent processes the user’s chat input, queries the Notion database for relevant records, and generates a summary or response based on the retrieved data. It uses a combination of OpenAI and Anthropic models to produce accurate and concise outputs.
  + **Configuration:**
    - **Input:** The chat message provided by the user.
    - **System Message:** Includes instructions to query the Notion database and format the response in a user-friendly way, including markdown-formatted URLs.
    - **Output:** JSON representing the AI agent’s response.

5. Search Notion Database

* **Node:** *Search notion database*
  + **Type:** HTTP Request (LangChain Tool HTTP Request)
  + **Description:**  
    This node sends a POST request to the Notion API to search within the specified database. The search query is dynamically constructed based on keywords and tags provided by the user.
  + **Configuration:**
    - **URL:** Constructed using the Notion database ID.
    - **Request Body:** JSON object with filters (by keyword and tag) and sort parameters.
    - **Placeholder Definitions:** For keyword and tag values.

6. Search Inside Database Record

* **Node:** *Search inside database record*
  + **Type:** HTTP Request (LangChain Tool HTTP Request)
  + **Description:**  
    This node retrieves detailed content from a specific Notion page by using its page ID. It extracts structured content (such as text from paragraphs, headings, and list items) from the page.
  + **Configuration:**
    - **URL:** Uses a placeholder {page\_id} replaced with the actual page ID from search results.
    - **Fields:** Specific fields like paragraph text, headings, and list items.
    - **Optimization:** Configured to optimize response for easier processing.

7. Window Buffer Memory

* **Node:** *Window Buffer Memory*
  + **Type:** Memory Buffer Window (LangChain Memory Node)
  + **Description:**  
    This node stores recent interactions (chat messages) in a buffer to maintain context during the conversation. It ensures that the AI agent has access to recent chat history for more coherent responses.
  + **Configuration:**
    - **Context Window Length:** Configured to store a specified number of recent messages.

8. Additional UI Components

* **Sticky Notes:**  
  Several sticky note nodes are used throughout the workflow to provide documentation and instructions directly within the n8n canvas. They serve as inline guides for users to understand the setup, configuration, and usage of the workflow.
  + **Sticky Note:** Overview of the workflow purpose.
  + **Sticky Note1:** FAQ and troubleshooting tips.
  + **Sticky Note2:** Template setup video link.

9. Output

* **Return Success:**  
  Once the workflow generates the modified n8n workflow template (which includes updated query components for the specific Notion database schema), the final output is returned as a JSON object. This JSON can be directly imported into n8n as a new workflow.

**Workflow Data Flow**

1. **User Input:**  
   The workflow is triggered by a chat message containing a Notion database URL.
2. **Database Retrieval:**  
   The Notion node fetches the database details using the provided URL.
3. **Data Formatting:**  
   The input data is standardized using the Format schema node, preparing it for the AI agent.
4. **AI Processing:**  
   The AI agent uses the formatted input and performs queries on the Notion database, searching for relevant records and summarizing the findings.
5. **Search Operations:**
   * The "Search notion database" node sends a query to the Notion API.
   * The "Search inside database record" node retrieves detailed content from individual pages.
6. **Context Management:**  
   The Window Buffer Memory node retains recent chat context to improve response coherence.
7. **Final Output:**  
   The generated workflow template (modified for the new Notion database schema) is output as a JSON object that the user can copy and import into their n8n instance.

**Setup Instructions**

1. **Notion Credential:**
   * Create and configure a Notion integration.
   * Share the target Notion database with the integration.
   * Add the Notion integration credential to your n8n workspace.
2. **OpenAI Credential:**
   * Set up an OpenAI API key and add it to n8n.
   * Configure the OpenAI Chat Model node with the appropriate model (e.g., gpt-4o).
3. **Chat Trigger:**
   * Configure the chat trigger node to listen for incoming messages.
   * Ensure the webhook URL is publicly accessible.
4. **Test the Workflow:**
   * Send a test message containing a valid Notion database URL.
   * Verify that the workflow retrieves the database details, processes the data through the AI agent, and returns a valid JSON workflow template.
5. **Review and Customize:**
   * Review the generated workflow template.
   * Customize the template further if needed to match your specific use case.

**Final Output**

The final output is a JSON representation of a fully configured n8n workflow template that you can import into your n8n instance. This template is specifically tailored for querying the provided Notion database and integrating with an AI assistant for knowledge retrieval.