**AI Agent to chat with you Search Console Data,**

using OpenAI and Postgres

What does this do?

* **Chat Interface for Data Insights:**  
  It starts with a webhook that receives chat messages (the “chatInput”) from users, along with a unique session ID and message date. This forms the basis of the conversation.
* **Storing Conversation History:**  
  A Postgres Chat Memory node logs your conversation history into a database table (insights\_chat\_histories), so the AI agent can reference past interactions and keep context.
* **AI-Powered Analysis:**  
  An AI agent, powered by OpenAI’s GPT-4o model, processes your chat input. It’s configured with a detailed system prompt that instructs it to:
  + Retrieve the list of connected Search Console properties.
  + Ask clarifying questions to understand exactly which property and data you want to analyze.
  + Construct a custom JSON API request based on your natural language input (for example, setting date ranges, dimensions, and row limits).
* **Constructing the API Call:**  
  A “Set fields - Construct API CALL” node extracts and sets parameters (like request type, start and end dates, dimensions, and property URL) from the input JSON, preparing a call to the Search Console API.
* **Tool Calling for Data Retrieval:**  
  The AI agent then uses a dedicated tool workflow (called “SearchConsoleRequestTool”) to make the API request:
  + If you want **custom insights**, it calls the Search Console API to fetch data based on your specified criteria.
  + If you ask for a **list of properties**, it retrieves the accessible website list.
* **Processing and Aggregation:**  
  The workflow aggregates the API responses (using array aggregation nodes) and converts them into a structured format—like a markdown table—so the data is easy to read.
* **Responding Back to You:**  
  Finally, the processed insights are sent back as a response to your initial chat message, completing the interactive cycle.