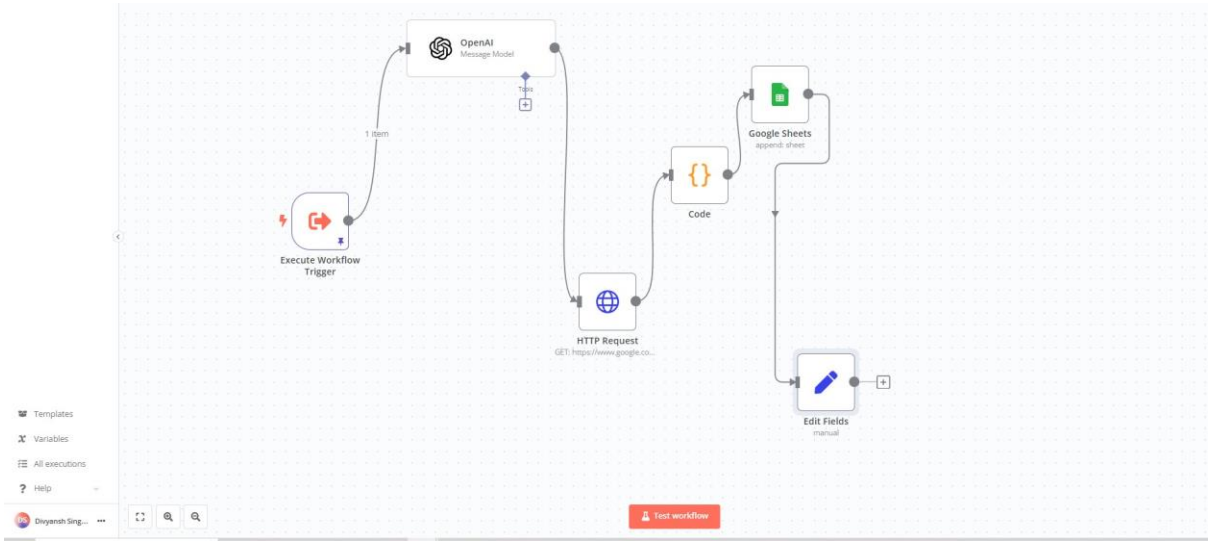
Step-by-Step Guide to Create a LinkedIn URL Scraper in n8n



Create the Scraper Workflow

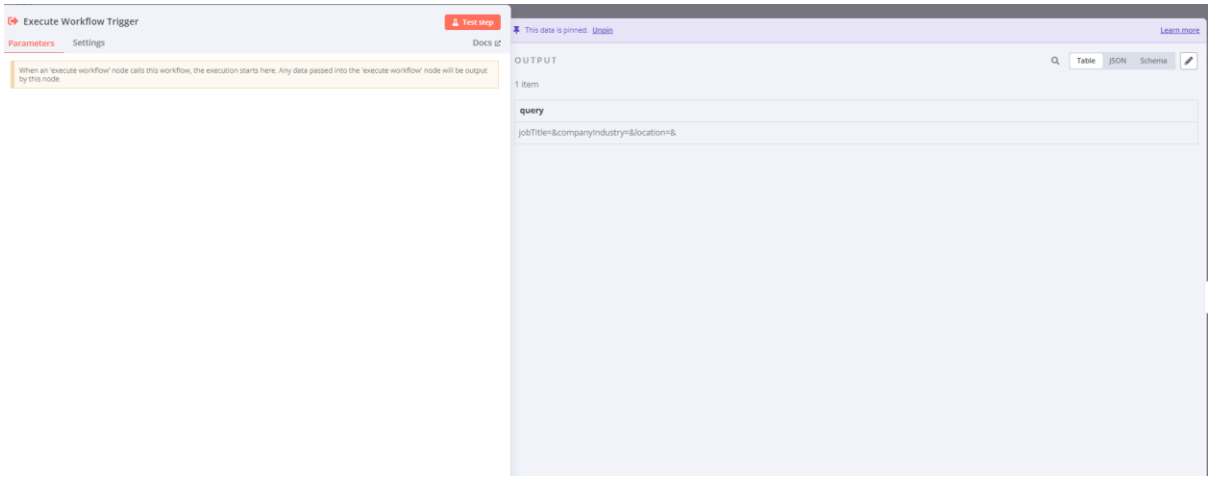
1.1. Set Up Trigger

- Add a **Trigger Node** that is called by another workflow.

1.2. Define Input Parameters

- Create an input structure (JSON) for the workflow:

```
{
  "jobTitle": "",
  "companyIndustry": "",
  "location": ""
}
```



1.3. Configure OpenAI Node

- Add an **OpenAI Node**.
    - Select the model (e.g., **GPT-4**).
    - Set the type to **System Prompt**.
    - In the message field, type:

      Parse the JSON 'query' and output the following parameters separately:

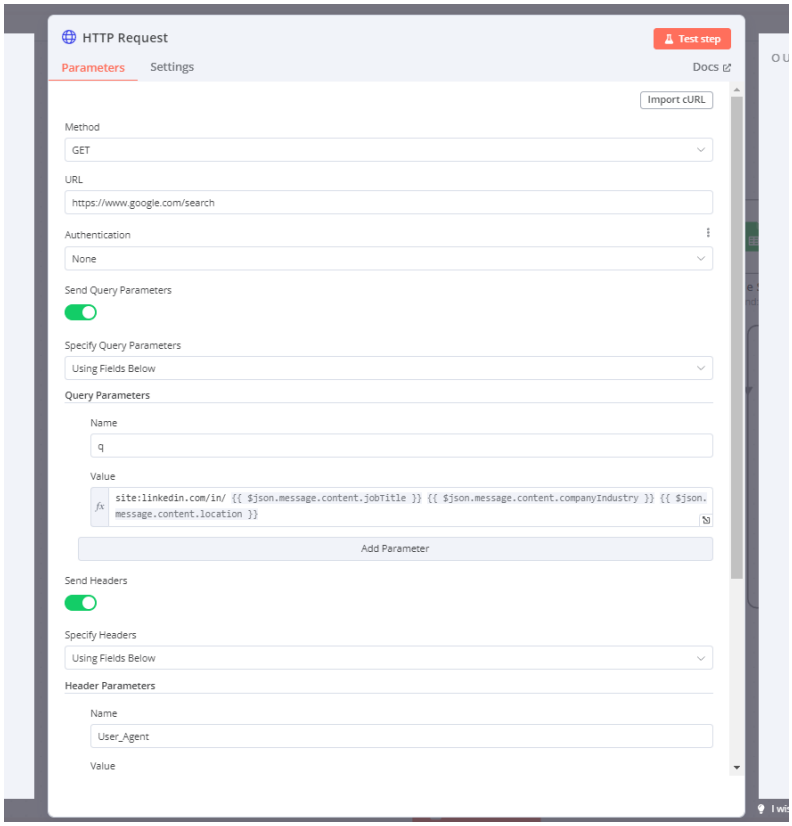
      jobTitle

      companyIndustry

      location

      Use query as the input from the previous step.

1.4. Set Up HTTP Request Node



- Add an **HTTP Request Node**.
    - Set the method to **GET**.
    - In the **Query Parameters**, define:
        - **Name**: q
        - **Value**: site:linkedin.com/in/ {{ $json.message.content.jobTitle }} {{ $json.message.content.companyIndustry }} {{ $json.message.content.location }}
    - In the **Headers** section, add:
        - **Name**: User-Agent
        - **Value**: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/115.0.0.0 Safari/537.36
    - This User-Agent header helps to mimic a real browser, which can improve the chances of successful scraping.

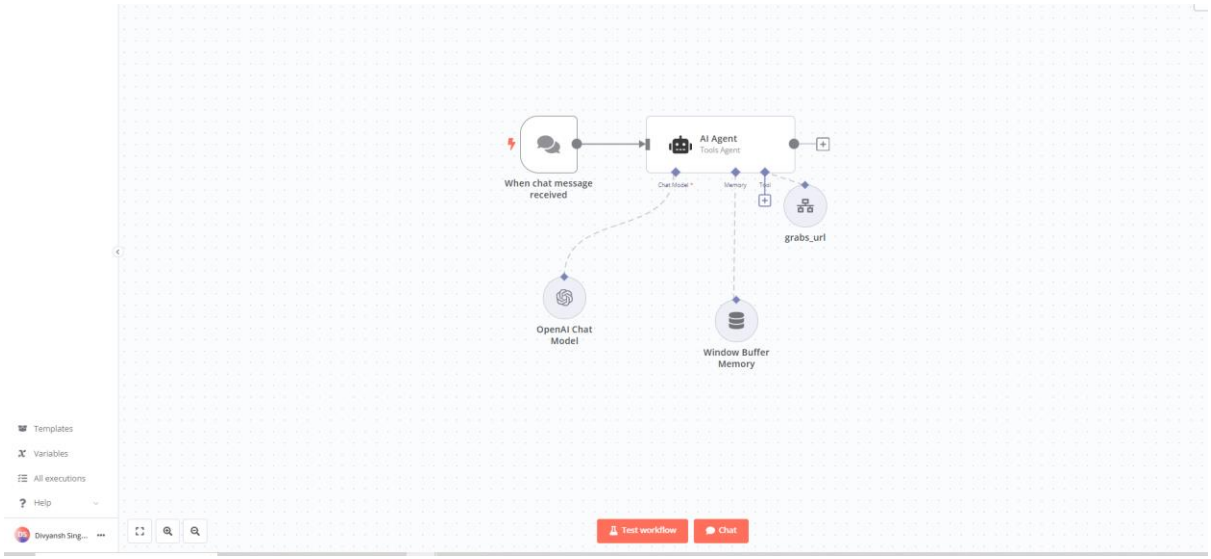
1.5. Configure Code Node for Parsing HTML

- Add a **Code Node** with JavaScript to parse the response from the HTTP request. You should have existing code for parsing the LinkedIn URLs from the HTML response. If it's already in your repository, simply copy it into this node.

1.6. Configure Google Sheets Node

- Add a **Google Sheets Node** to append data.

- Configure OAuth client for Google Sheets access.
- Select the target sheet where the URLs will be stored.
- Edit the node to set the response field to indicate completion:
  - **Field**: response
  - **Value**: done



Create the Agent Workflow

2.1. Set Up the Chat Agent

- Create a new workflow for the agent.
- Add a **Chat Message Received Node** to trigger when a chat message is received.

2.2. Add OpenAI Agent Node

- Add an **OpenAI Node** to manage the chat.
  - Select **GPT-4** for the chat model.

    - Use default settings for buffer memory.

2.3. Connect to Scraper Tool

- Add a node to **Grab URL**.
    - Select your previous LinkedIn URL scraper tool as the action.

2.4. Save and Test the Agent Workflow

- Click **Save** on your workflow.
- Test by sending a query through the chat (for example: “Find CEOs in real estate in Chicago”).

3. Final Testing and Validation

- After saving, run tests by entering different queries into the agent chat to ensure the scraping tool returns the expected LinkedIn URLs.

Conclusion

By following these updated steps, you will have included the necessary User-Agent header in your HTTP request to ensure better performance when scraping LinkedIn URLs. If you need any more adjustments or help, just let me know!