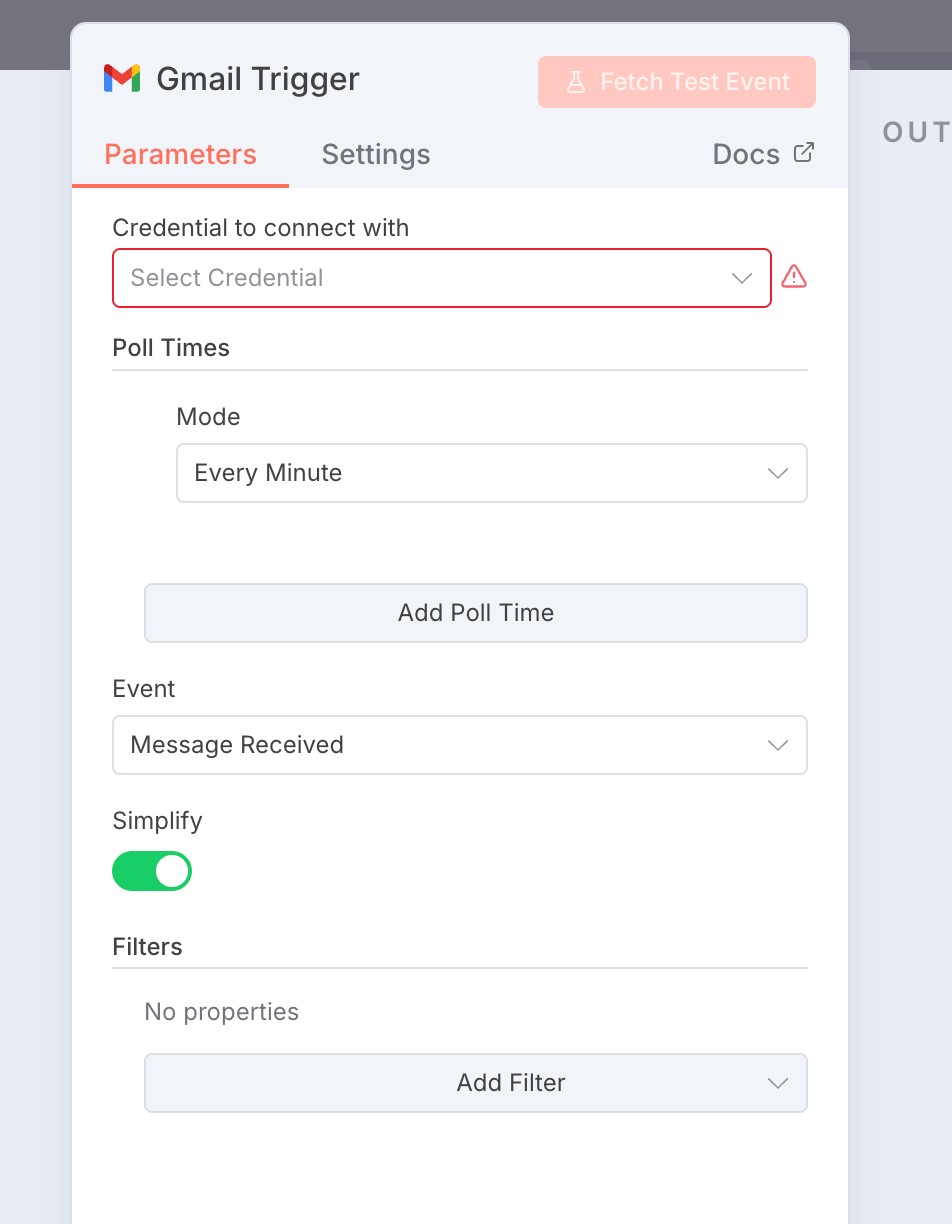
Docker

Creating a workflow :

1. From gmail option under trigger choose on message received you will get a pop up below



1. Go to <https://console.cloud.google.com/> and complete your sign up
2. In [console.google.com](http://console.google.com/) —> click on create new project then click on new project in the dialog box prompted

**🔹 Step 1: Enable Gmail API in Google Cloud**

1. Go to Google Cloud Console.
2. Select your project (or create one).
3. Navigate to **APIs & Services → Library**.
4. Search for **Gmail API** and click **Enable**.

**🔹 Step 2: Create Credentials**

1. Go to **APIs & Services → Credentials**.
2. Click **+ Create Credentials → OAuth client ID** (or Service Account if required by your workflow tool).
3. Configure:
   * **Application type**: choose *Web application* or *Desktop* depending on your workflow tool.
   * Add **Authorized redirect URIs** (your workflow tool usually gives you this).
4. Save → you’ll get a **Client ID** and **Client Secret**.

**🔹 Step 3: Link Gmail API to Workflow Tool**

This depends on the workflow platform (Zapier, n8n, Power Automate, or custom). But generally:

* Go to the **Gmail trigger** block in your workflow.
* It will ask you to **connect an account** or **add credentials**.
* Choose **OAuth 2.0** (sign in with Google).
* Paste the **Client ID** and **Client Secret** (if asked).
* Grant permissions (scopes like gmail.readonly, gmail.modify, gmail.send).

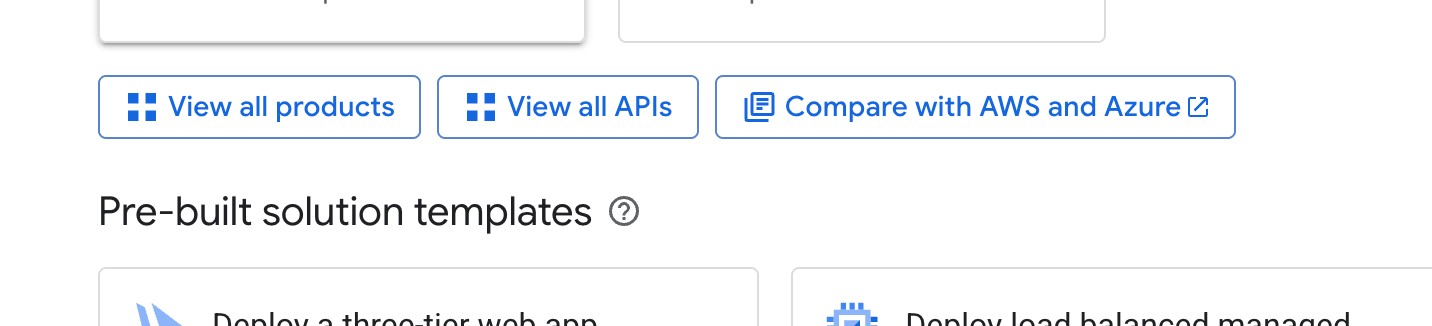
**🔹 Step 4: Test the Connection**

* Send yourself a test email.
* Your workflow should trigger on **“New Email Received”**.
* If your workflow needs actions (like reply, save to Sheets, forward), those use the same Gmail API connection.

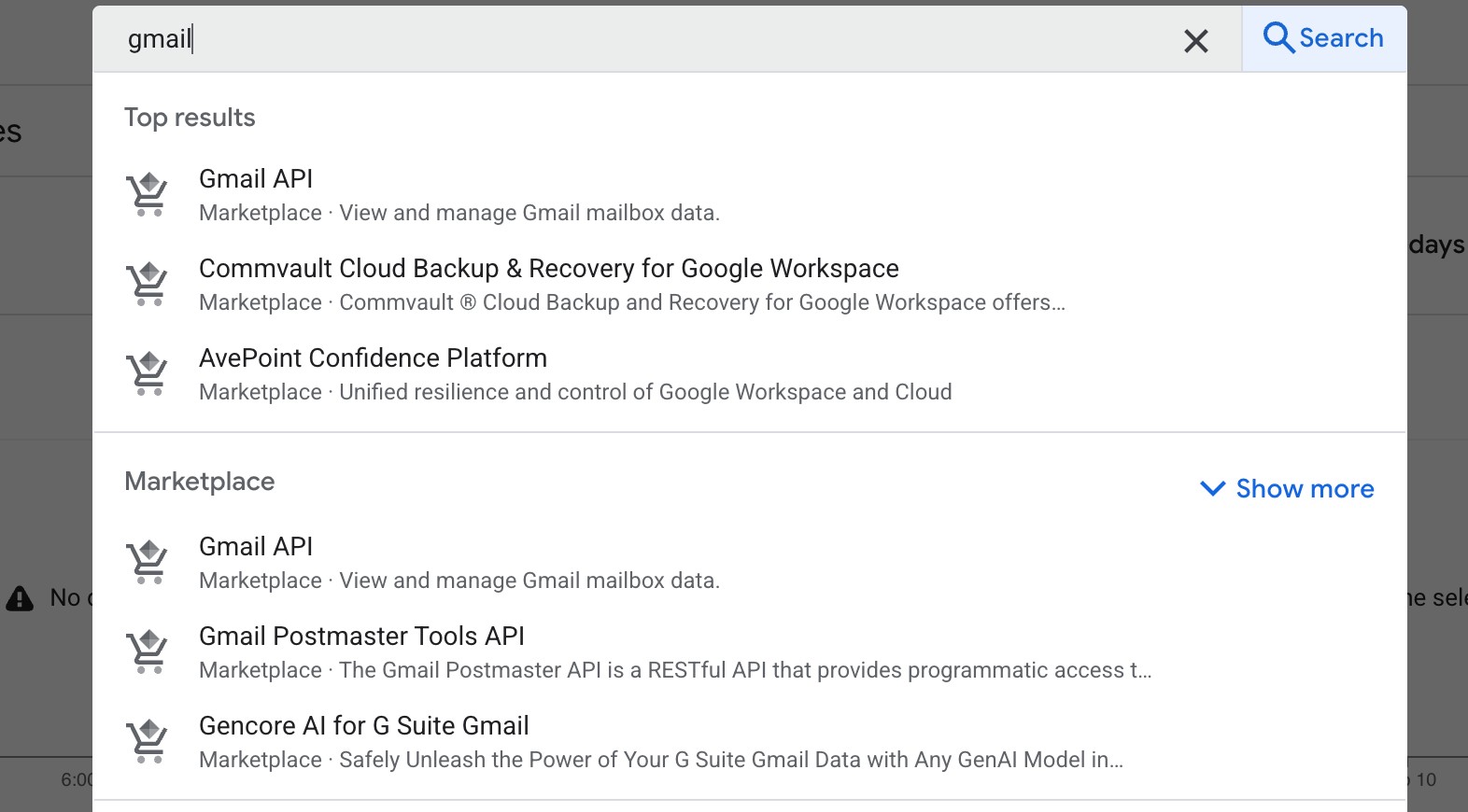
✅ In short:

* Google Cloud → Enable Gmail API + create credentials.
* Workflow tool → Add Gmail trigger → Link using those credentials.
* Test → Emails now start the workflow.

1. Go to your project and go to view all API’s and go API’s and Services at top left



1. Go to gmail API and click on ENABLE



**🔹 Steps to Get Client ID & Secret**

1. **Go to Credentials**
   * In Google Cloud Console, navigate to **APIs & Services → Credentials**.
2. **Create OAuth Client ID**
   * Click **+ Create Credentials → OAuth Client ID**.
   * In the dropdown:
     + If your workflow tool (like **n8n**) gives you a **redirect URL**, select **Web application**.
     + If not, and you’re just testing, you can select **Desktop app**.
3. **Configure Application**
   * **Name**: give it something like n8n Gmail OAuth.
   * **Authorized redirect URIs**:
     + Paste the redirect URL your workflow tool shows you (for n8n it’s usually something like:
     + https://<your-n8n-domain>/rest/oauth2-credential/callback

or http://localhost:5678/rest/oauth2-credential/callback if running locally).

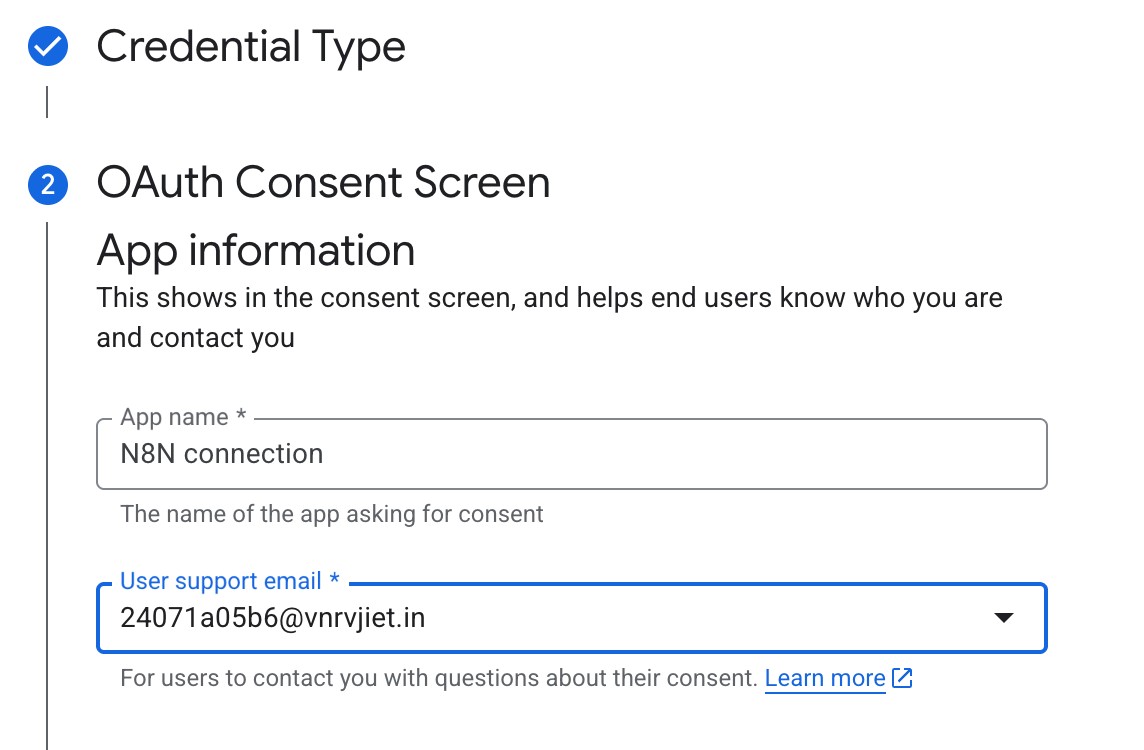
1. **Save**
   * Once you hit **Create**, Google will generate two things:
     + **Client ID**
     + **Client Secret**
2. **Copy These Values**
   * They’ll be shown immediately in a popup.
   * You can also find them later under **Credentials → OAuth 2.0 Client IDs** (click your client name).

**🔹 Next Step**

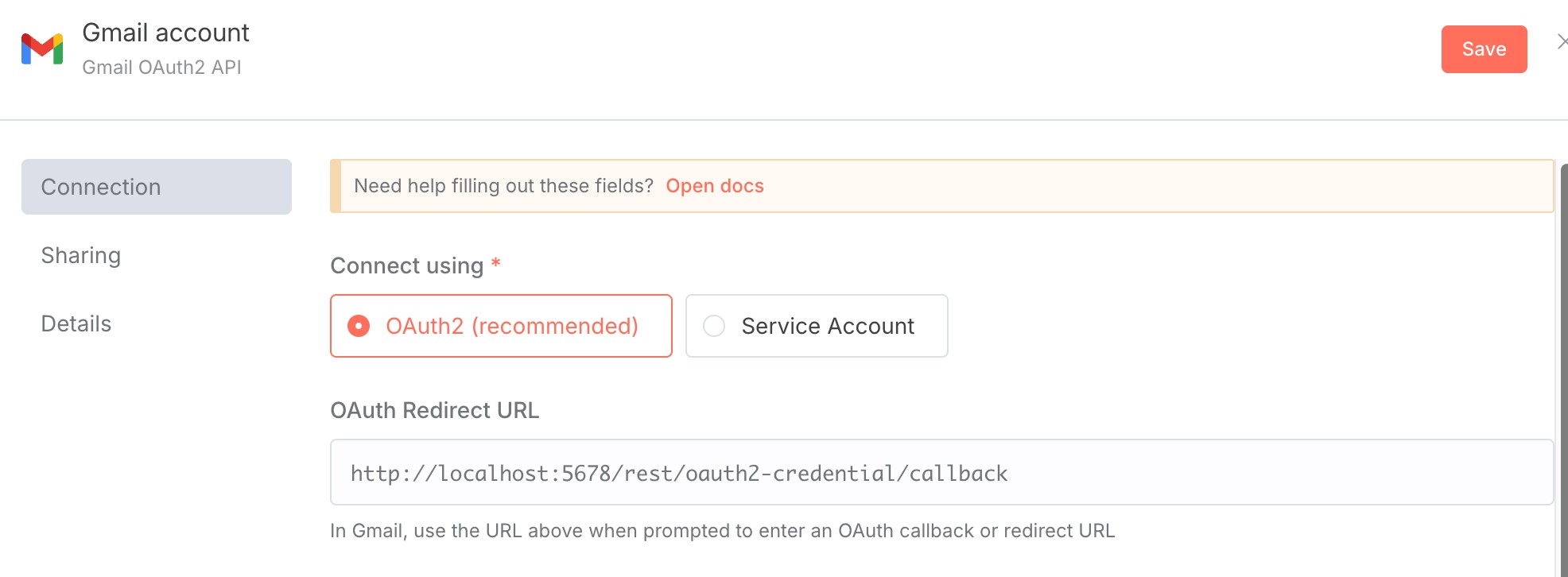
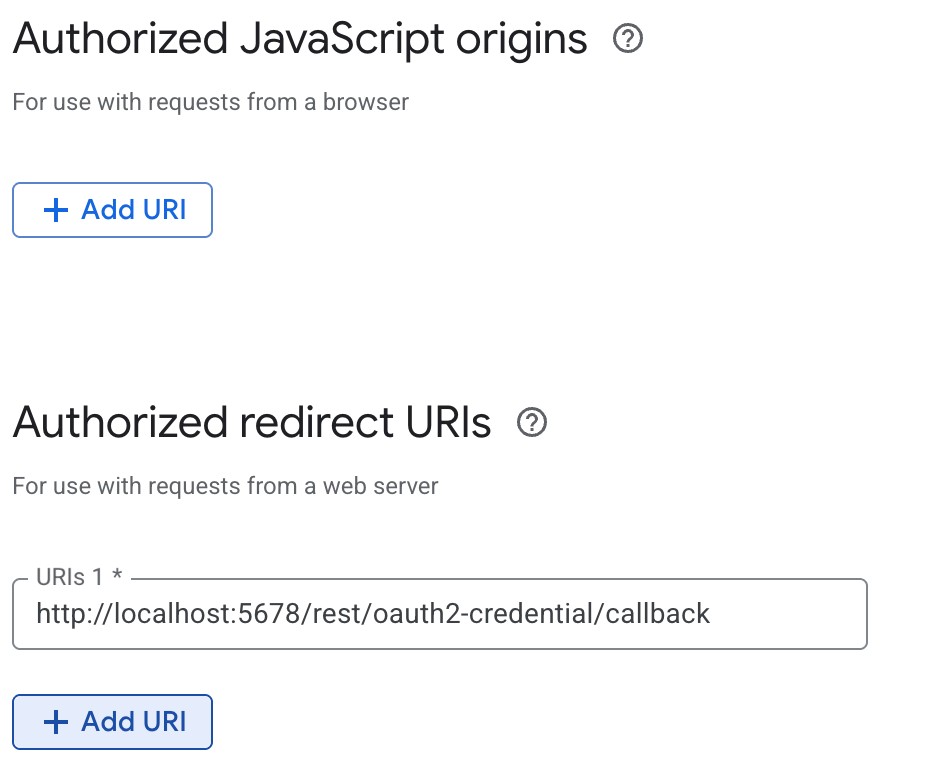
Go to your workflow tool (n8n, Zapier, etc.):

* Open the Gmail node → Add new credentials → Select **OAuth2**.
* Paste in the **Client ID** and **Client Secret**.
* Paste the same **redirect URI** you used in Google Cloud.
* Authenticate → Login with your Gmail account → Approve access.

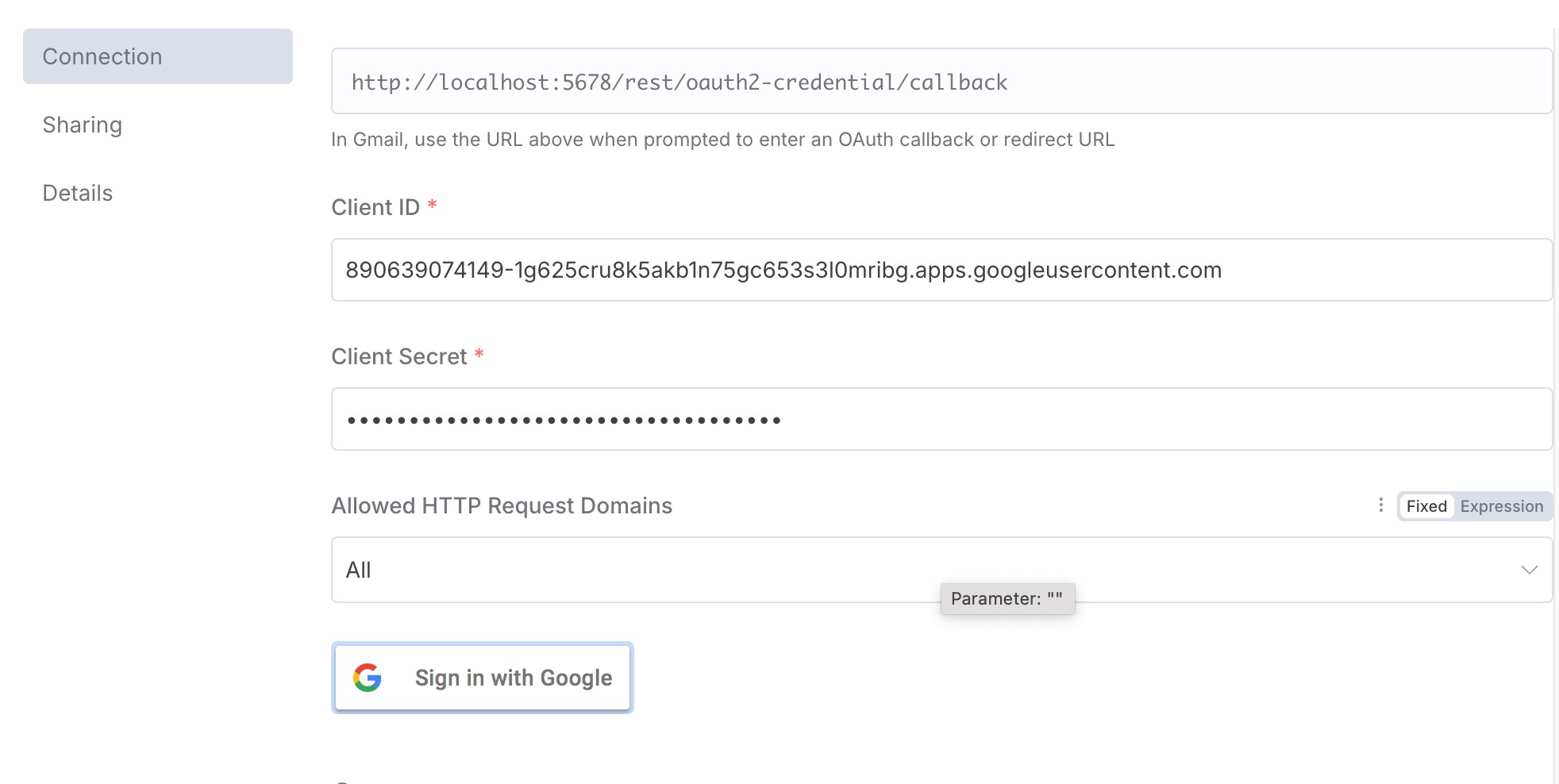
1. Click on create credentials



1. In the gmail trigger —> create new credential copy paste the redirect URL

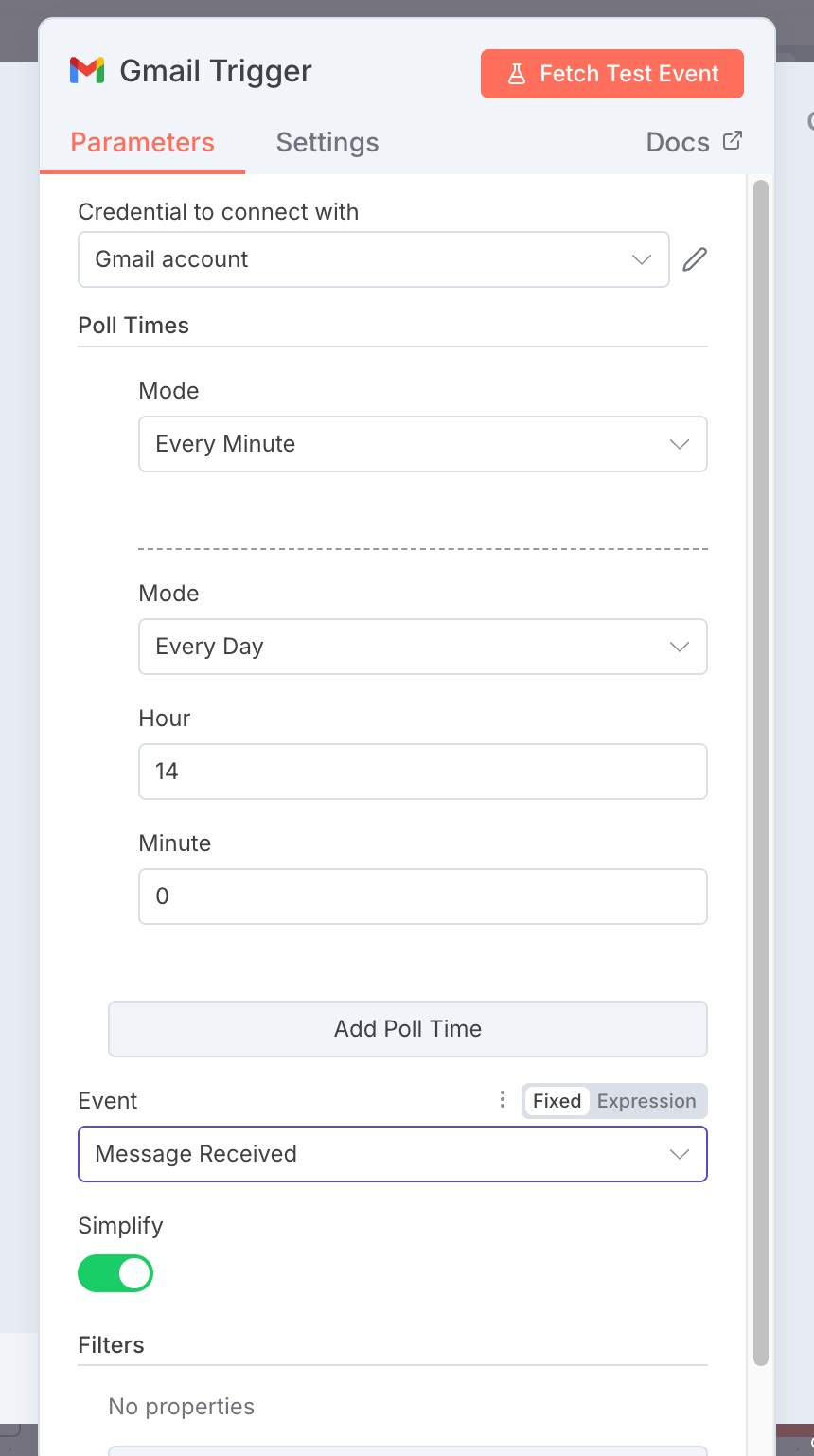


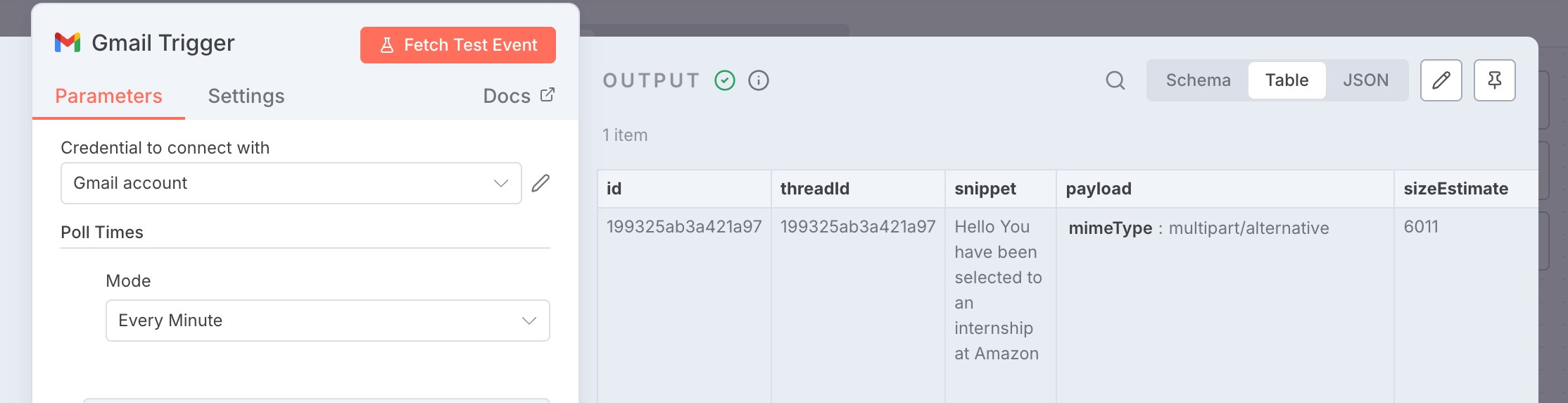
1. Go to google auth platorm then clients check whether the client is created
2. Sign in with google



1. If there is an issue with sign in then check with the port number of local host <http://localhost:55000/workflow/new?projectId=NEht528hjKzGTRMY>
2. To change the port number from 55000 to 5678 run the below command in the terminal

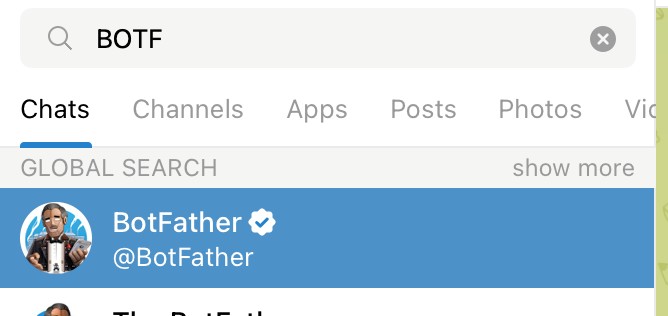
docker run -p 5678:5678 n8nio/n8n

1. Fill this as follows : 
2. Then send an email to the email with docker account
3. Then click on fetch text event to get the output



For telegram :

1. Go to telegram —> search for BOTFather then click on start or type “/start”



1. Create new workflow and search for telegram and select Send a Message option
2. Paste the created bot token in access in new credential

Use this token to access the HTTP API:

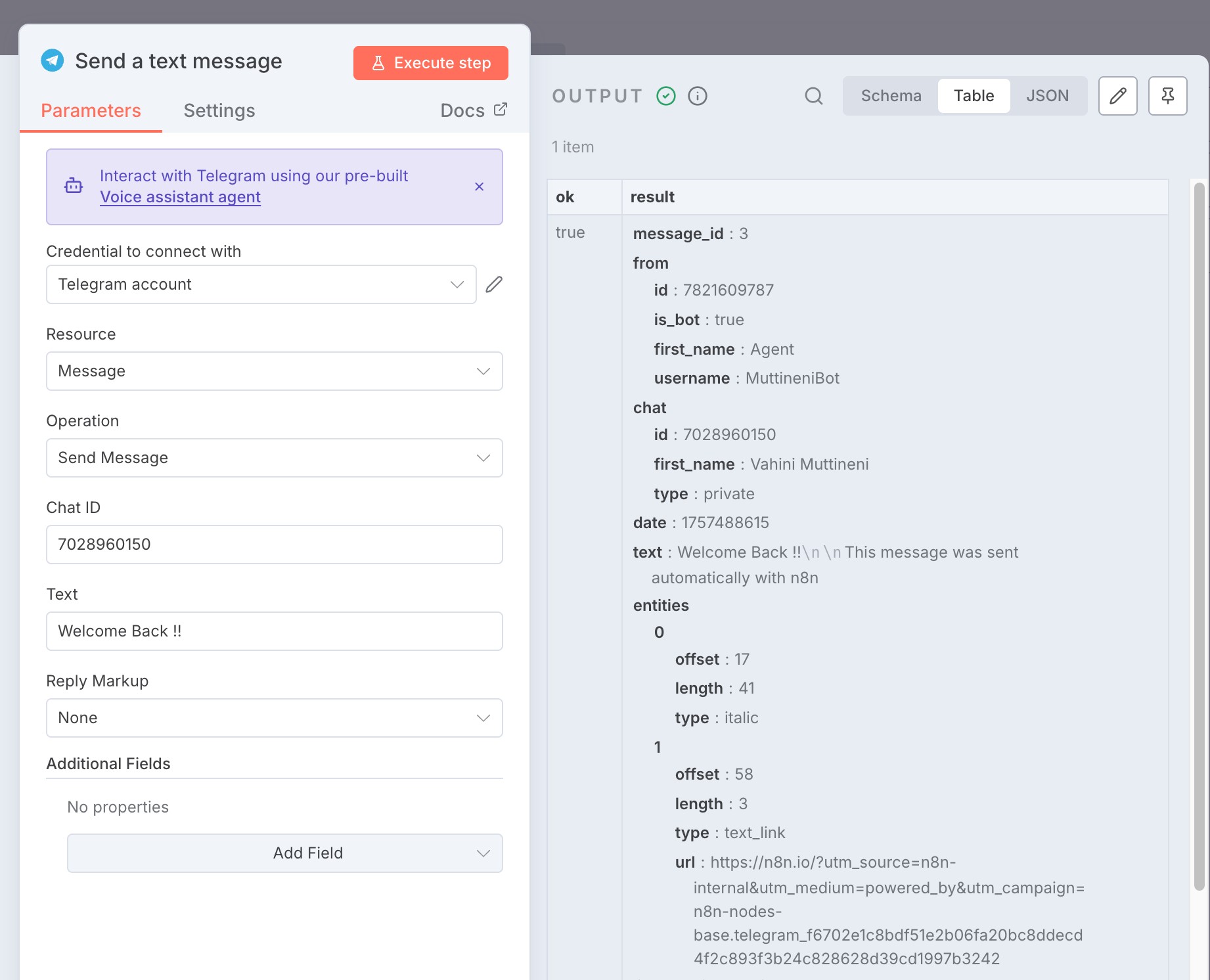
7821609787:AAGfG09HSSupwVD3XT3R9HVngJKTAE9VHT8

1. Chat with your bot to get the chat if in the browser

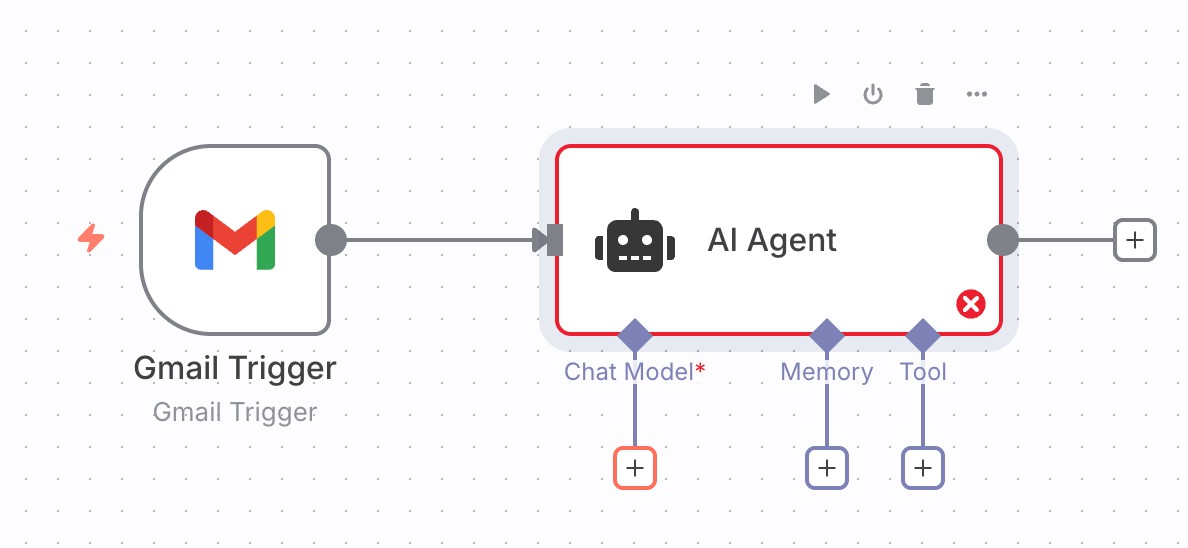
To open in browser use the below link [https://api.telegram.org/bot<token>/getUpdates](https://api.telegram.org/bot%3ctoken%3e/getUpdates)

5. Paste the chat id n8n

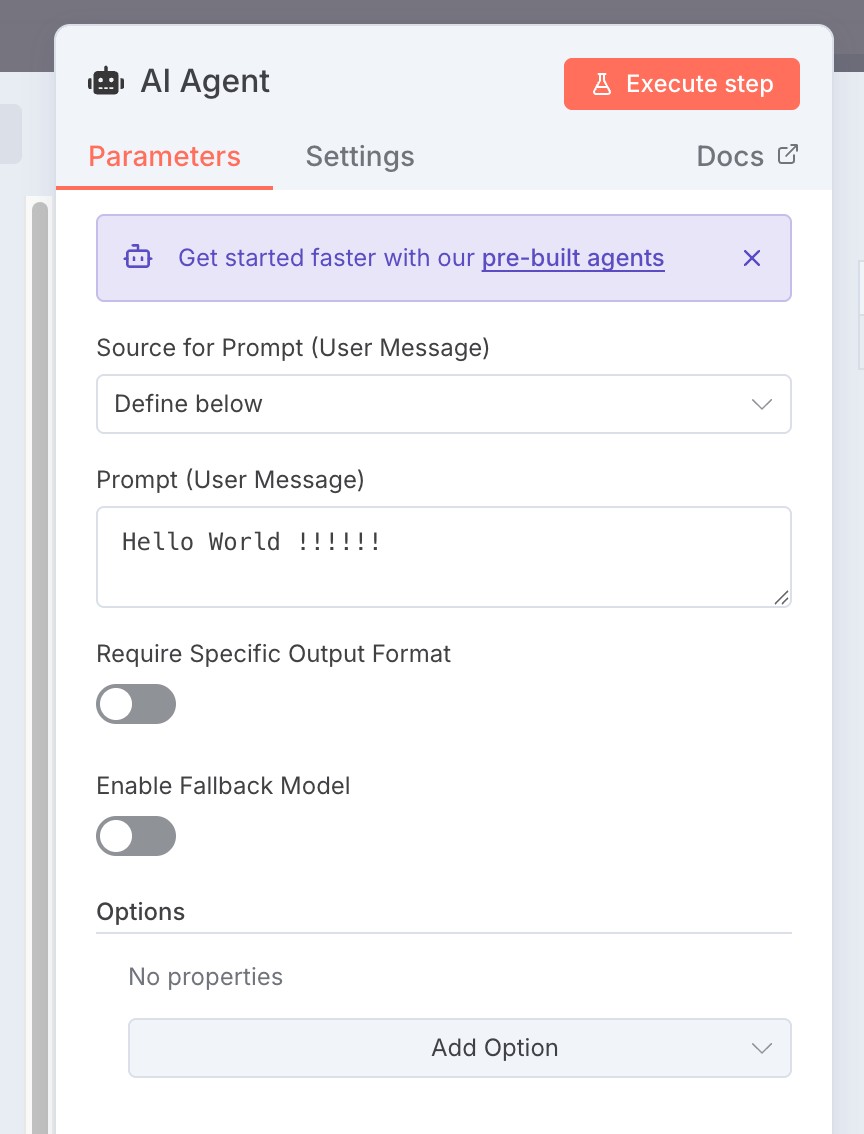
6. Execute the step :



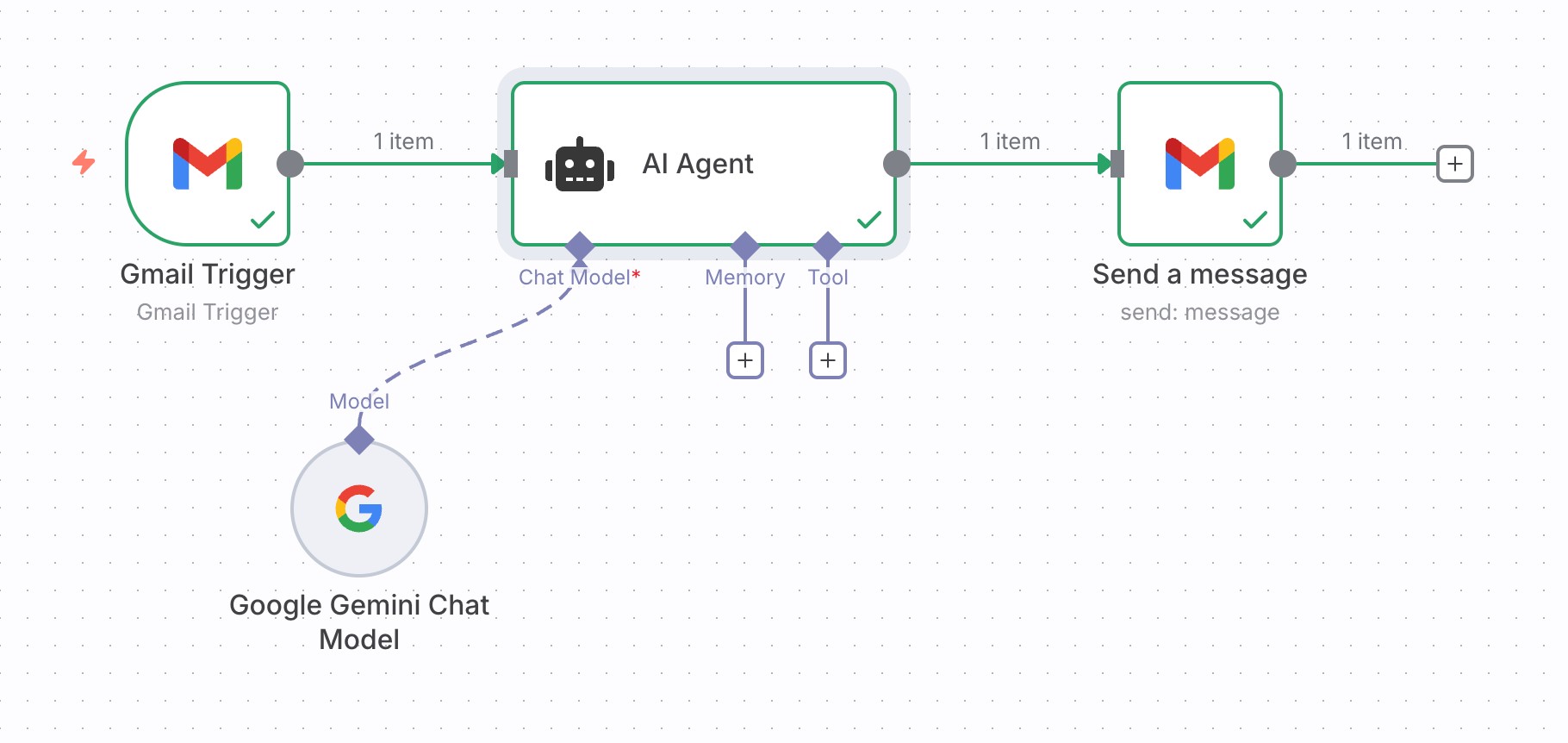
1. Open Google AI Studio
2. Navigate to API keys
3. Then click on create new API key
4. Go to gmail testing working created before
5. Then search for AI agent



1. Click on + at Chat Model add Google Gemini Chat Model. Here you should add API key while creating new credential
2. When you want to test the step created now first execute the step which is created before i.e Gmail Trigger



1. Set the options in AI agent as above
2. Set the workflow as below :



1. Execute each and every step then check your email whether you got any mail or not .

Results of the SESSION held on 10/09/2025

