Voice Agent

Hey there! 👋 Meet your new **Voice Agent**, your friendly assistant right inside Telegram. Just send a voice or text message, and it'll understand you, chat with you, and even help you find information. Whether you want to ask questions, get recommendations, or just have a casual conversation, it's like talking to a human—but faster, smarter, and always available.

Powered by AI, it remembers your recent chats so the conversation feels natural, and it can even access information from the web when you need it. Think of it as your personal companion on Telegram, ready to make your day a little easier—and a lot more fun! Let me guild you in buliding your own voice agent!

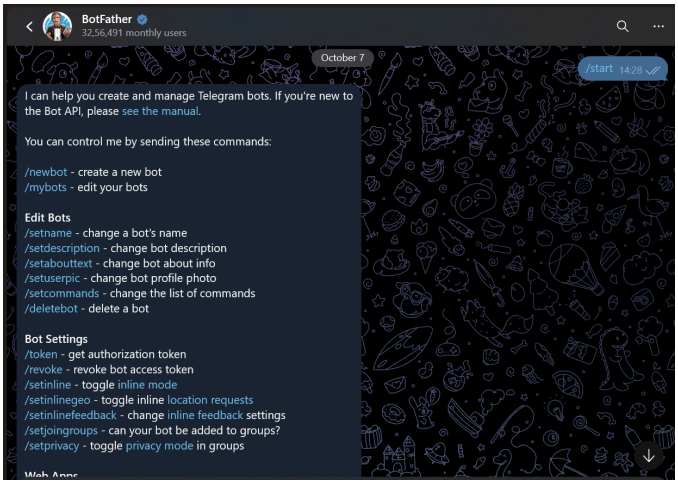
Create Telegram Bot

Before we get into n8n, we first need to create our Telegram bot — this is what will actually talk to you. Telegram makes this part really easy with a built-in tool called **BotFather**.

Think of BotFather as the “parent” of all Telegram bots. Every bot you see on Telegram was created through it.

Just click this link: **@BotFather**. It’ll open directly in Telegram.

Once you’re there, you’ll see a list of commands.

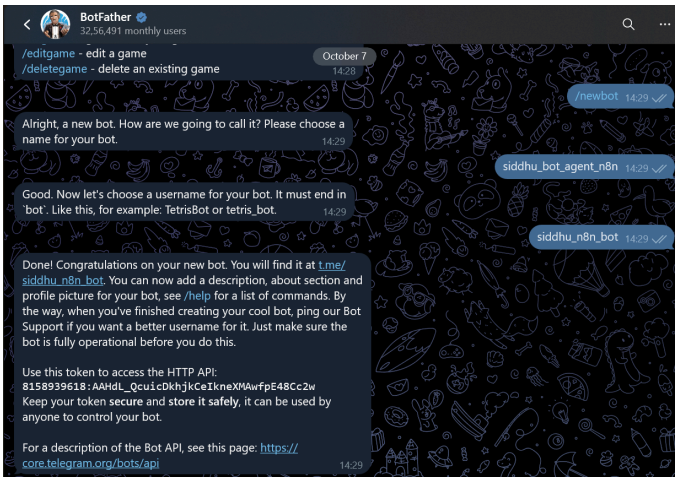


All you need to do is type:

```
\newbot
```

BotFather will then ask you for two things:

1. A **name** for your bot — this is the display name people will see.
2. A **username** — this must end with “bot” (something like `myassistant_bot` or `harshhelper_bot`).



After that, BotFather will instantly create your bot and send you a message with a **unique API token**.

Copy that token and keep it somewhere safe — we'll need it soon when connecting your bot to n8n.

Telegram Trigger Node

Alright, now that your bot is alive, let's help it *do* something. To make that happen, we'll connect it to **n8n** — the tool that'll power all the automation behind it.

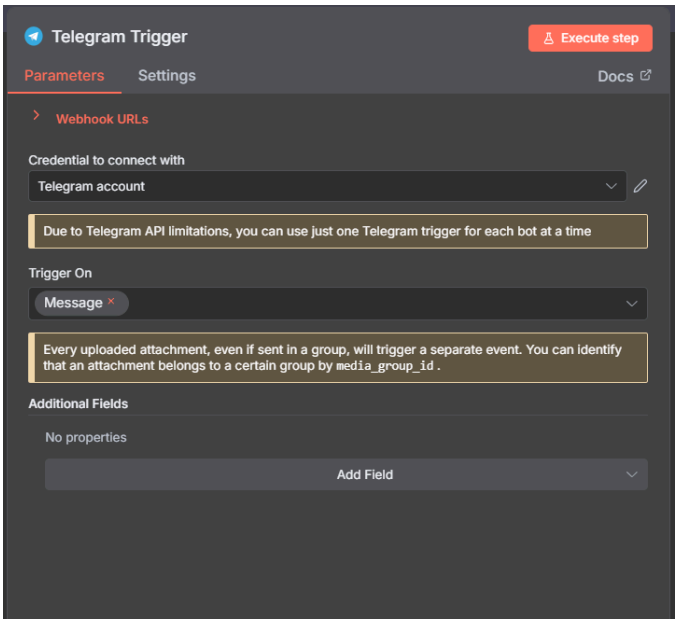
Open up your **n8n dashboard** and start a new workflow. You'll see a big empty canvas — that's where all your logic will go.

Now, click the **"+"** button and search for **Telegram**. From the options, choose **Telegram Trigger** — this node basically listens for messages sent to your bot.

Once you add it, n8n will ask you to connect your Telegram account. This is where you'll need the **API token** you got from **BotFather** earlier.

Just paste that token in the "Access Token" field and hit **Connect**.

Give it a second, and boom — your Telegram bot and n8n are now officially talking to each other



Get a File Node

Once your bot is set up, it's time to see it in action! 🎙️ Send a voice message to your Telegram bot, and let it capture what you said.

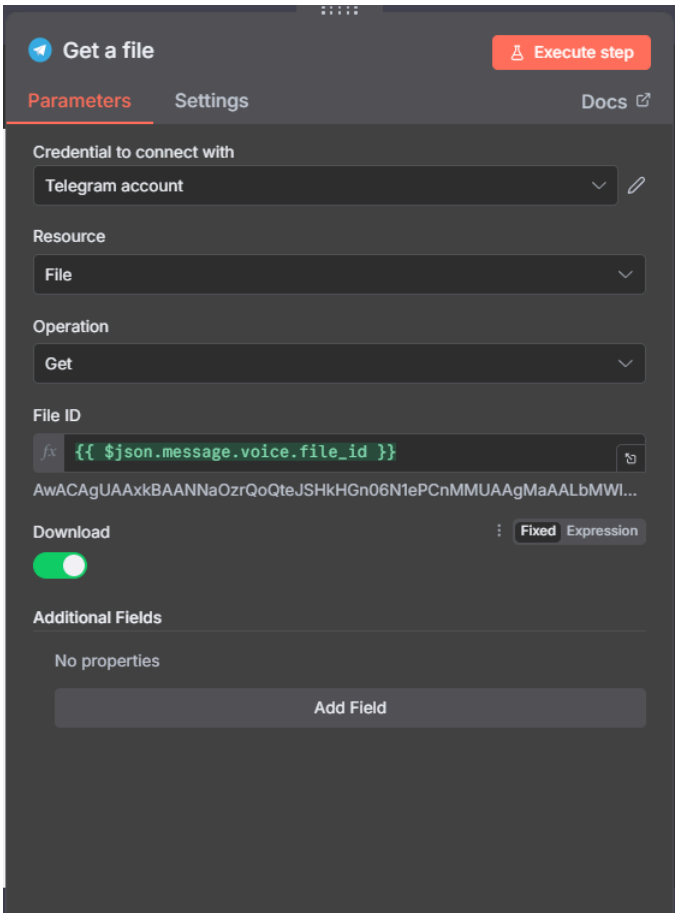
Next, head back to **n8n**. We want to grab that voice message so we can do something with it — maybe transcribe it, analyze it, or store it. Add a new **Telegram node** and choose the **"Get File"** operation.

For the **File ID**, use:

```
{{ $json.message.voice.file_id }}
```

This tells n8n exactly which voice message to fetch. Once the workflow runs, n8n will download the voice file for you — ready to process however you like!

It's as simple as talking to your bot and letting n8n handle the heavy lifting in the background. Your voice is now digital and ready to do all sorts of things. 🎧💬



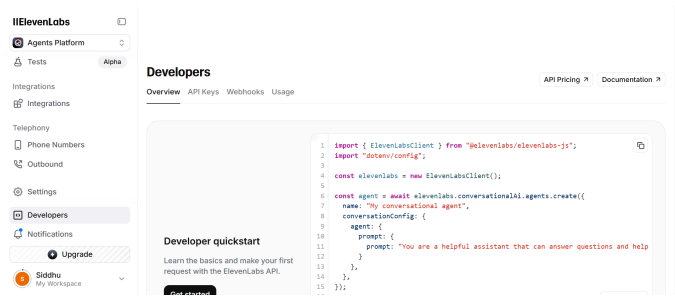
Alright, now that we can capture voice messages, it's time to make our bot *smarter*. To do that, we'll use **ElevenLabs**, which can process voice and turn it into text, or even generate voice responses.

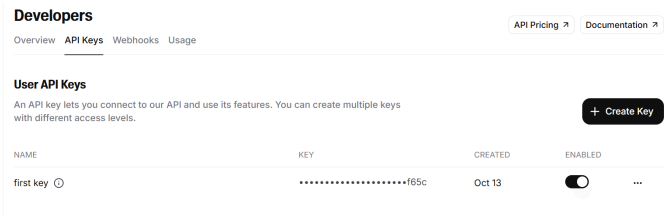
Eleven Labs Api

First, head over to **elevenlabs.io** and create an account if you don't have one yet. Once you're logged in, go to the **Developers** section — this is where the magic begins.

Here, you'll find an option to generate an **API key**. Think of this key as your bot's passport to ElevenLabs' AI powers. Copy it somewhere safe, because we'll need it in the next steps to connect your Telegram bot workflow to ElevenLabs.

With this setup, your bot will be able to process and respond to voice messages in a truly human-like way. 🎙️✨





Elevenlabs Node

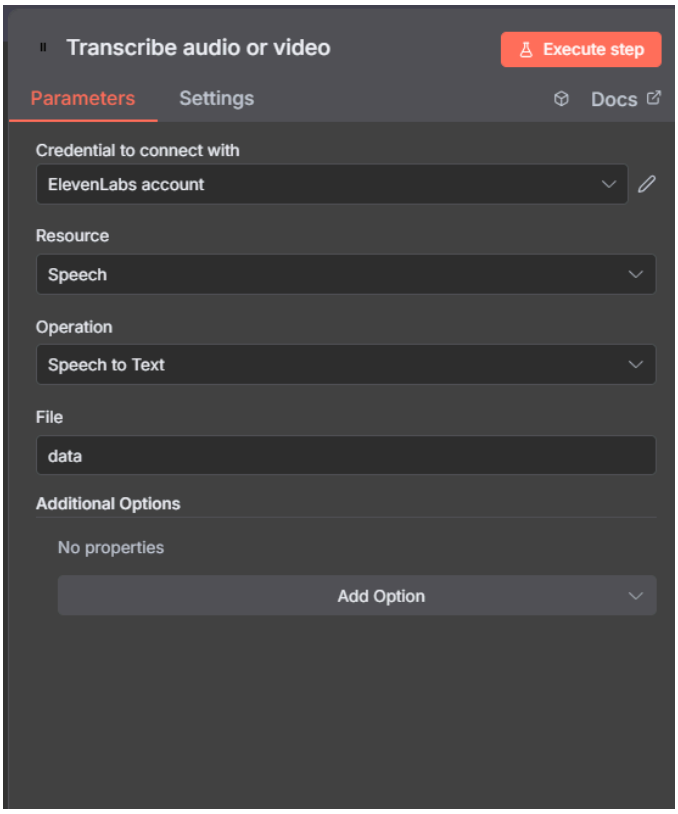
Now that we have our **ElevenLabs API key**, it's time to connect it to n8n and start turning those voice messages into text.

Back in your **n8n workflow**, click the **"+"** button and search for **ElevenLabs**. Select the **"Transcribe Speech to Text"** node — this is where your bot will listen to the voice file and convert it into text.

In the node settings:

- **Credential:** Choose your **ElevenLabs account** (using the API key you just created).
- **Resource:** Select **Speech**.
- **Operation:** Pick **Speech to Text**.
- **File:** Set this to the voice file captured earlier (for example, `data` from your Telegram node).
- You can also explore **Additional Options** to tweak the transcription settings if needed.

Once set up, every voice message sent to your Telegram bot will automatically be transcribed into text. 📝💬 Your bot can now understand what users are saying — the first step toward a fully interactive conversation!



AI Agent node

With the voice message now transcribed, it's time to make your bot *actually talk back* using AI. In n8n, add the **AI Agent node** to your workflow. This node will take the text from the voice message and generate a smart, human-like response.

Here's how to set it up:

- **Source for Prompt (User Message):** Select **Define below**.
- **Prompt (User Message):** Use

```
{{ $json.text }}
```

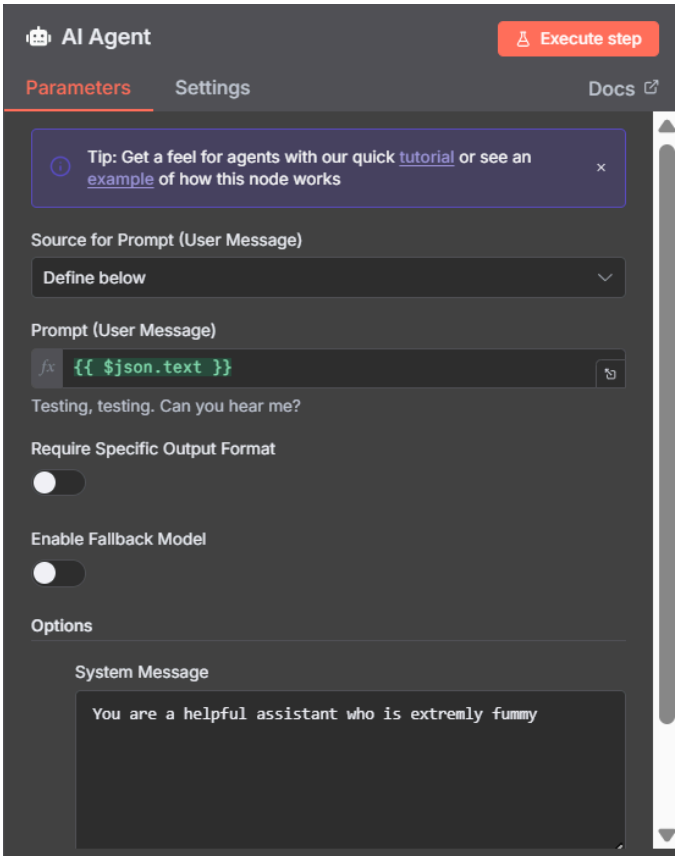
This ensures your AI agent gets the exact text from the transcribed voice message.

- **System Message:** Here's where you can define your bot's personality. For example:

> You are a helpful assistant who is extremely funny.

You can also enable options like **Require Specific Output Format** or **Fallback Model** if you want more control over how the bot responds.

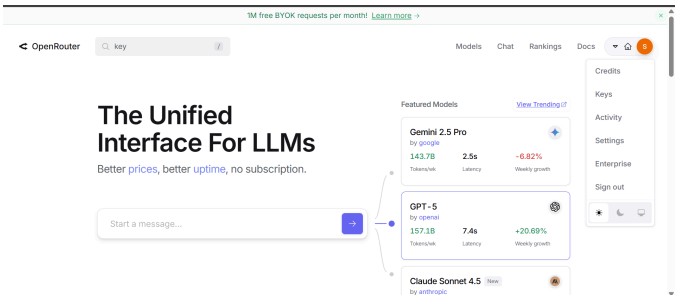
Once this node is configured, your Telegram bot can read the user's message, think like a human, and reply in a natural, engaging way — making conversations feel lively and fun. 🤖💬



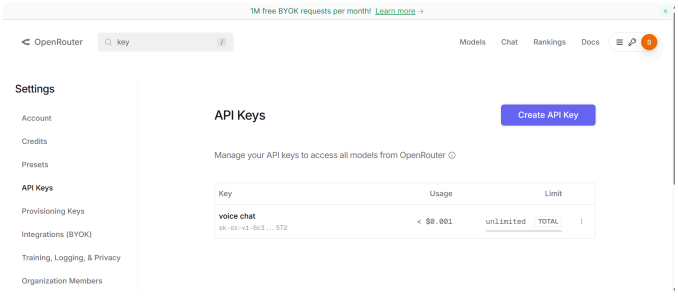
OpenRouter Node

Next, we'll hook our bot up with a powerful **AI chat model** using **OpenRouter**. This will allow your bot to generate smart, human-like replies.

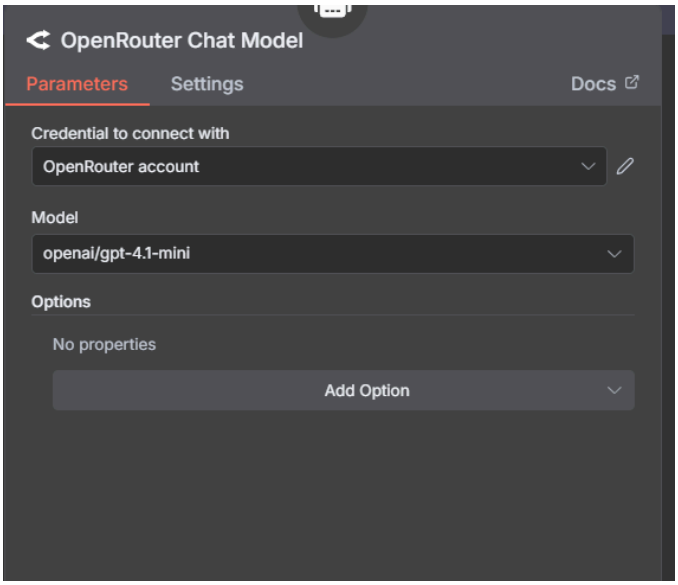
First, head over to the **OpenRouter Chat Model** page and create an account if you don't already have one. Once you're logged in, look at the top-right corner of the page — click on your **account** icon. You'll see a list of options; choose **API Keys**.



From there, create a new key and copy it somewhere safe. This key is what lets your n8n workflow talk to the OpenRouter AI model.



Finally, go back to **n8n** and paste this API key into your AI Agent node credentials. With this setup, your bot is now ready to understand user messages and generate intelligent, natural responses.



Simple Memory Node

To make conversations feel more natural, we'll add a **Simple Memory node** in n8n. This allows your bot to **remember recent messages** so it can carry context from one message to the next — like a real conversation.

Add the **Simple Memory node** after your AI Agent node in the workflow. You don't need to worry about complex setups here; it just keeps track of a set number of recent messages. Later, this memory can be used by the AI node to generate responses that are context-aware, making the chat feel more human and engaging.

Convert Text to Speech

Now that our AI has generated a reply, it's time to make your bot **talk back** — literally! To do this, we'll use ElevenLabs again, but this time with the **"Convert Text to Speech"** node in n8n.

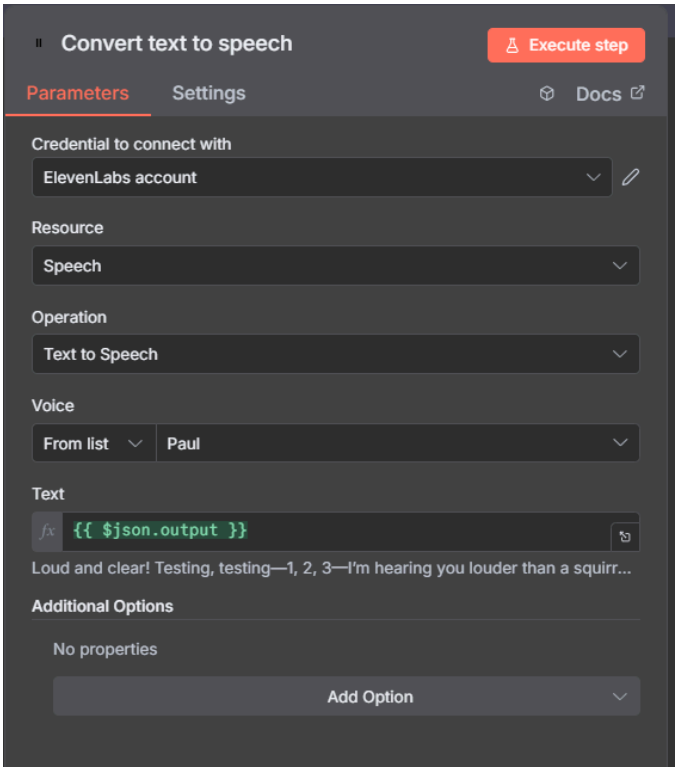
Add the node to your workflow and set it up like this:

- **Credential:** Select your **ElevenLabs account**.
- **Resource:** Choose **Speech**.
- **Operation:** Pick **Text to Speech**.
- **Voice:** Select from the list (for example, **Paul**).
- **Text:** Use

```
{{ $json.output }}
```

This ensures that the text generated by the AI Agent is converted into a voice message. You can also explore **Additional Options** if you want to adjust things like pitch, speed, or style.

Once this is set up, every AI response will be transformed into a voice message — ready to send back to the user on Telegram. Imagine sending a question and hearing your bot respond with a natural, lively voice!



Send an Audio File Node

Hurray — the final step!

We've come a long way — from recording a voice message, transcribing it, running it through an AI agent, and converting the reply back into speech. Now it's time to send that voice message back to the user on Telegram!

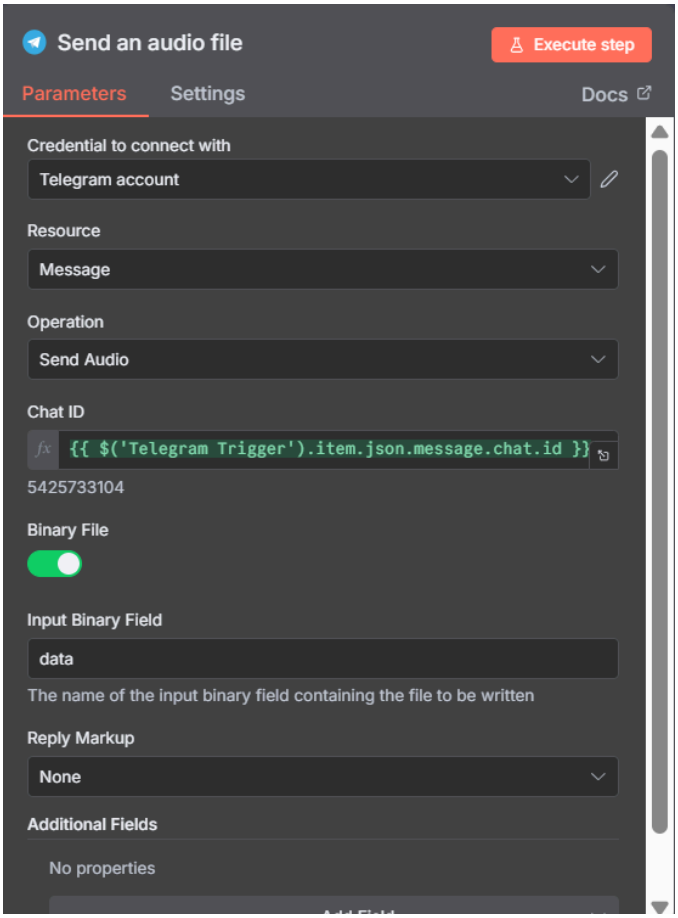
In your **n8n workflow**, add one last **Telegram node** and set it up as follows:

- **Credential:** Connect your **Telegram account**.
- **Resource:** Select **Message**.
- **Operation:** Choose **Send Audio**.
- **Chat ID:** Use

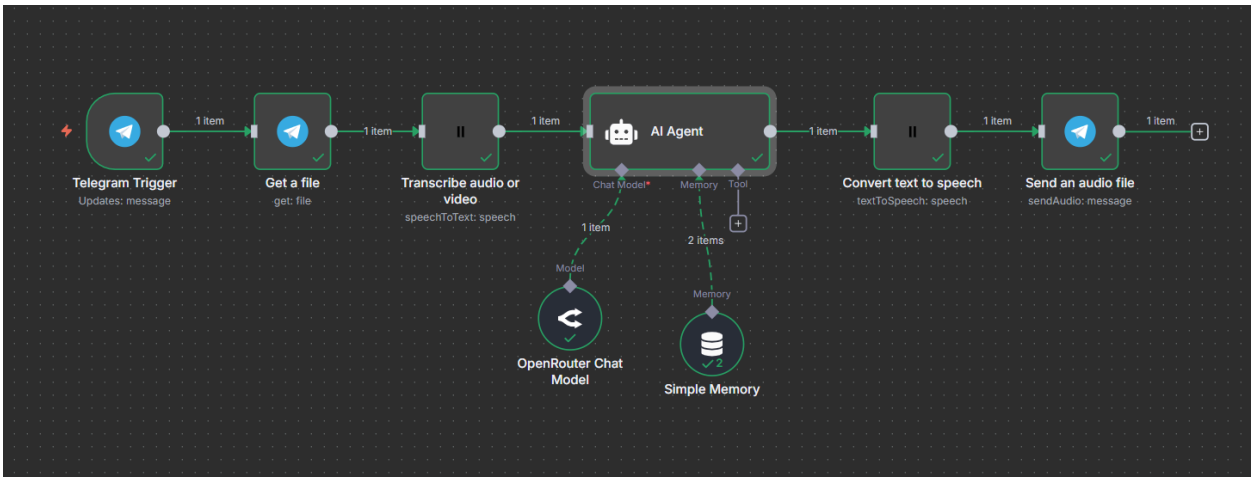
  ```
  {{ $('Telegram Trigger').item.json.message.chat.id }}
  ```

  This ensures the voice reply goes directly to the same user who sent the message.

- **Binary File:** Set to **Input Binary Field → data** (this is where the voice file from ElevenLabs is stored).
- **Reply Markup:** Leave it as **None** for now, unless you want to add buttons or menus later.



Final workflow



Conclusion

And that's it — your Telegram Voice Agent is ready! It can now listen, think, and talk back like a real person. Using **n8n**, **ElevenLabs**, and **OpenRouter**, you've built a simple but powerful AI chatbot that feels alive.

This is just the start — you can keep adding new voices, fun personalities, or smarter memory to make it even better. Go ahead, test it out, and enjoy chatting with your own AI-powered voice buddy!