

Sanjhai's Gmail Cleanup Assistant: User Guide

Sanjhai

July 2025

1 Introduction

Sanjhai's Gmail Cleanup Assistant is a free, open-source workflow built in **n8n**, an automation platform. This workflow helps you keep your Gmail inbox clean by automatically deleting emails from the following categories:

- **Spam:** Unwanted or junk emails.
- **Promotions:** Marketing and promotional emails.
- **Social:** Notifications from social media platforms.
- **Trash:** Emails in the trash folder.

The workflow runs daily at 2:00 AM and sends a summary report to your Telegram account, detailing the number of emails deleted in each category. This guide explains how to set up and use the workflow.

2 Prerequisites

Before using the workflow, ensure you have the following:

- An **n8n instance** (self-hosted or cloud-based). Download or sign up at <https://n8n.io/>.
- A **Gmail account** with API access enabled. You will need OAuth2 credentials from the Google Cloud Console.
- A **Telegram account** and a Telegram bot with a chat ID for receiving notifications.

3 Setup Instructions

Follow these steps to set up the workflow in n8n:

3.1 Download the Workflow

- Obtain the workflow JSON file (`Sanjhai_s_Gmail_Cleanup_Assistant.json`) from the source where it was shared (e.g., GitHub, Pastebin, or a direct download link).

3.2 Import the Workflow

- Open your n8n instance.
- Navigate to **Workflows > Import Workflow**.
- Upload the `Sanjhai_s_Gmail_Cleanup_Assistant.json` file or paste its contents.
- Save the workflow in n8n.

3.3 Configure Gmail API Credentials

- Go to the **Google Cloud Console** (<https://console.cloud.google.com/>).
- Create a new project or select an existing one.
- Enable the **Gmail API** under **APIs & Services > Library**.
- Set up OAuth2 credentials:
 - Go to **APIs & Services > Credentials**.
 - Create an **OAuth 2.0 Client ID** (select *Web application*).
 - Add <https://your-n8n-instance.com> to the **Authorized redirect URIs** (replace with your actual n8n instance URL).
 - Download the **client secret JSON** and note the **Client ID** and **Client Secret**.
- In n8n, go to **Credentials > Add Credential > Gmail OAuth2**.
- Enter the **Client ID** and **Client Secret**, then authenticate with your Gmail account.
- Save the credentials and assign them to the Gmail nodes (*Get Spam Emails, Get Promotional Emails, Get Social Emails, Get Trash Emails, Delete Spam Emails, Delete Promotional Emails, Delete Social Emails, Delete Trash Emails*).

3.4 Configure Telegram Bot

- Create a Telegram bot using **BotFather** (<https://core.telegram.org/bots#creating-a-new-bot>).
- Start a chat with **BotFather** on Telegram and send `/newbot`.
- Follow the prompts to name your bot and receive a **Bot Token**.
- Get your **Telegram Chat ID**:
 - Send a message to your bot or add it to a group.
 - Use a service like @GetIDsBot or check the bots updates via the Telegram API (<https://api.telegram.org/bot<YourBotToken>/getUpdates>).
- In n8n, go to **Credentials > Add Credential > Telegram API**.
- Enter your **Bot Token** and save.
- In the **Telegram Notification** node, update the `chatId` field with your **Telegram Chat ID**.

3.5 Test and Activate the Workflow

- Open the workflow in n8n.
- Manually run the workflow by clicking **Execute Workflow** to ensure it works as expected.
- Check the Telegram notification to verify the summary report.
- If successful, activate the workflow by toggling the **Active** switch. It will now run daily at 2:00 AM.

4 Usage Notes

- **Permanent Deletion:** The workflow permanently deletes emails, especially those in the trash folder. Test with a small `limit` (default is 250 emails per category) to avoid accidental data loss.

- **Gmail API Limits:** The Gmail API has quotas (e.g., 250 read/write operations per second). If you have a large inbox, you may need to adjust the `limit` parameter in the Gmail nodes or add delays to avoid hitting quotas.
- **Customization:** You can modify the schedule by changing the `triggerAtHour` in the *Schedule Trigger* node (default is 2:00 AM). Adjust the `limit` in the Gmail nodes if you want to process more or fewer emails.
- **Error Handling:** The workflow includes basic error handling to filter empty Gmail responses. Monitor the workflows execution in n8n for any issues.

5 Troubleshooting

- **Gmail Authentication Errors:** Ensure your OAuth2 credentials are correctly set up and that the Gmail API is enabled in Google Cloud Console.
- **Telegram Notification Issues:** Verify the **Bot Token** and **Chat ID** are correct. Test the bot by sending a manual message.
- **No Emails Processed:** Check the Gmail filters (e.g., `q: in:spam`) in the Gmail nodes. Ensure your inbox has emails in the specified categories.
- **API Quota Exceeded:** If you hit Gmail API limits, reduce the `limit` parameter or add a delay node between operations.
- For further help, contact Sanjhai at sanjhaiprakash18@gmail.com or visit the n8n Community Forum: <https://community.n8n.io/>.

6 License

This workflow is free to use, modify, and share under the **MIT License**. You can redistribute it as long as you include the original attribution to Sanjhai.

7 Contact

For questions, feedback, or assistance with setting up the workflow, please contact Sanjhai at sanjhaiprakash18@gmail.com. You can also join the n8n Community Forum (<https://community.n8n.io/>) or share your thoughts on X with the hashtag **#n8n**.