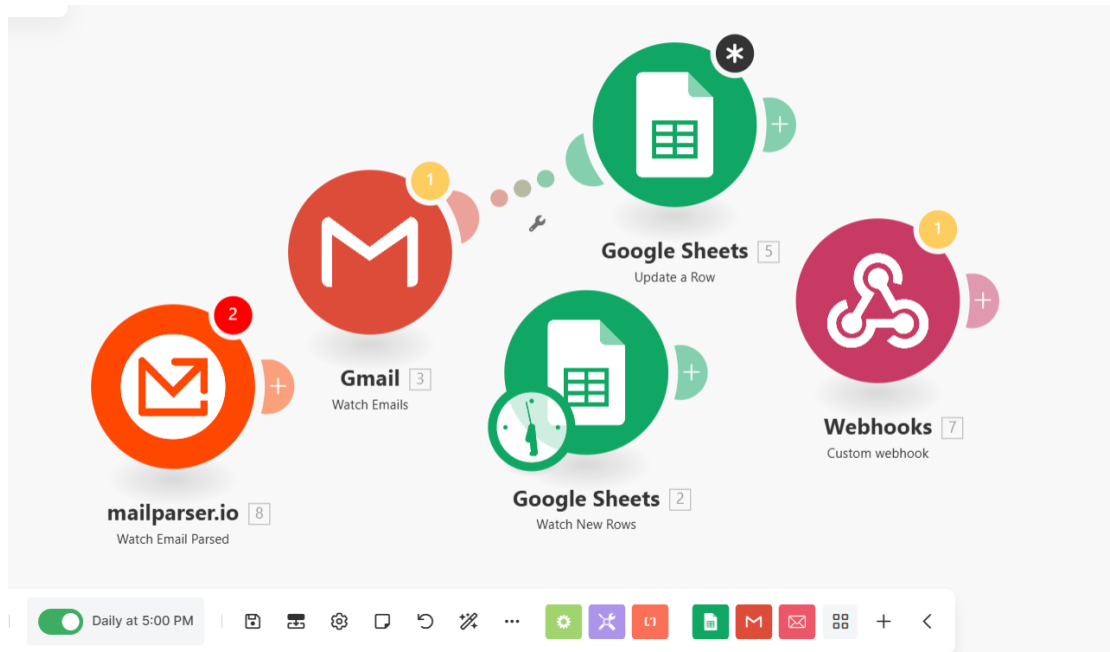# Email Campaign Automation Project Documentation

1. Overall Approach

My primary objective is to create an automated workflow that reads leads from a Google Sheet, sends personalized emails to these leads, updates their status in the sheet, and monitors for any replies. The workflow is built using Make and integrates with Google Sheets and an Gmail for sending emails. Webhook module is used to monitor incoming email replies and trigger notifications.

The process is divided into four main stages:

1. **Data Acquisition:** Reading lead data from a Google Sheet.

2. **Email Sending:** Sending personalized emails based on the data.

3. **Status Update:** Updating the lead's status and sent time in the Google Sheet.

4. **Reply Monitoring & Notification:** Capturing replies and updating the reply status or sending notifications.

---

2. Module Configurations

2.1 Google Sheets – Data Acquisition

Module: Google Sheets "Watch Rows"

Details:

- **Account Connection:** Connect to the Google account that owns the lead data sheet.

- **Spreadsheet Selection:** Choose the spreadsheet containing the mock data.

- **Worksheet Selection:** Select Sheet1 where the columns are:


  - name

  - Email

  - status

- sent time

- reply status

- **Trigger Condition:**

  - Configure the module to monitor new or updated rows.

  - Use a filter to only process rows where status is either empty or set to "Pending".

2.2 Email Sending

Module: Gmail

Details:

- **Account Connection:** Authorize Make to access my email account.

- **Recipient Address:** Dynamically map the Email field from the Google Sheet.
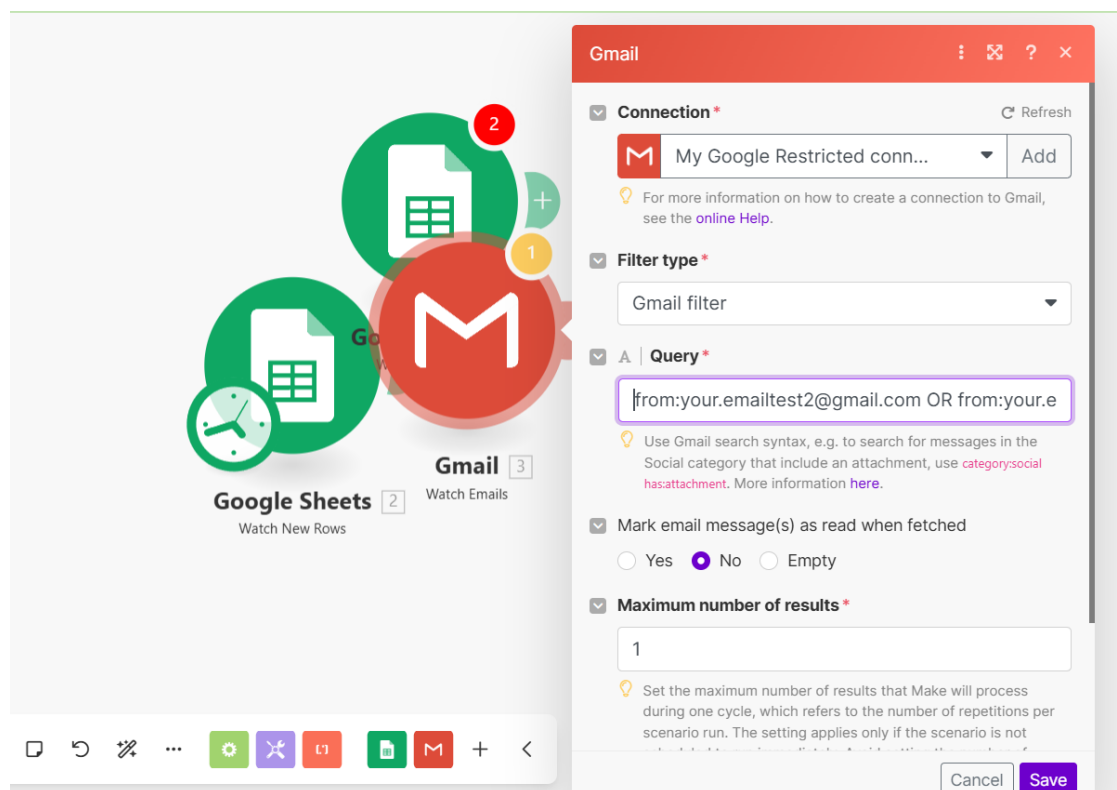
- **Query and Body:**

  - **Query: from:your.emailtest1@gmail.com OR from:your.emailtest2@gmail.com OR from:your.emailtest3@gmail.com**

  - **Subject example: Your Email is Invalid**

- **Body:** Create a custom message that may include HTML formatting and personalization using the data from the Google Sheet.

- **Dynamic Data Insertion:**

  - Use variables to inject the recipient's name and other relevant details into the email.
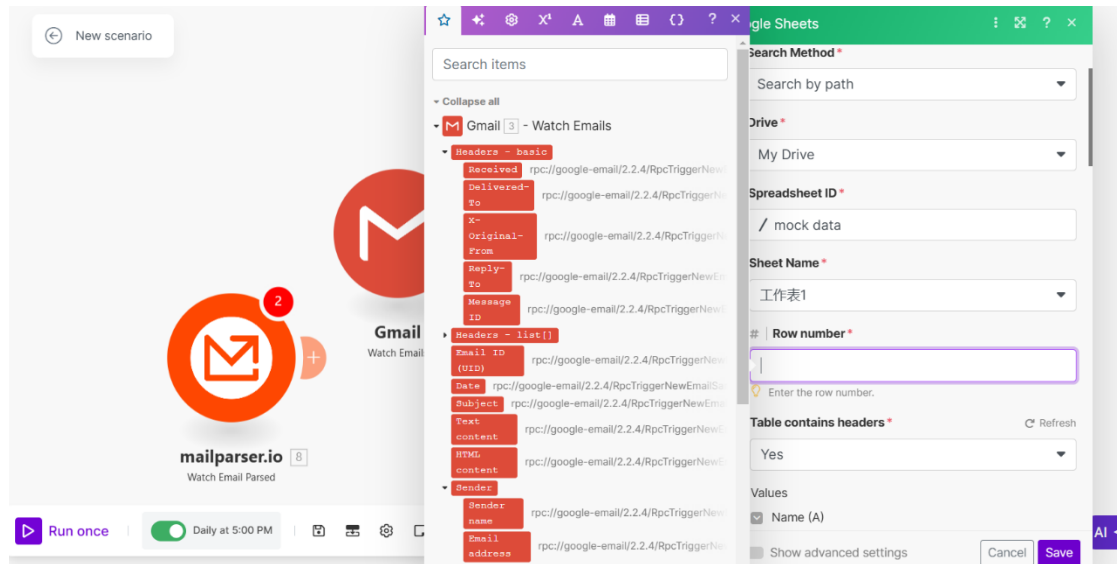
Error Handling in Email Module:

- **Retry Mechanism:** Configure automatic retries if the email fails to send (e.g., up to 3 retries with a 5-minute interval).

- **Fallback Path:** If retries fail, the workflow should trigger an error branch where the failure is logged, and the row's status can be updated to "Failed."
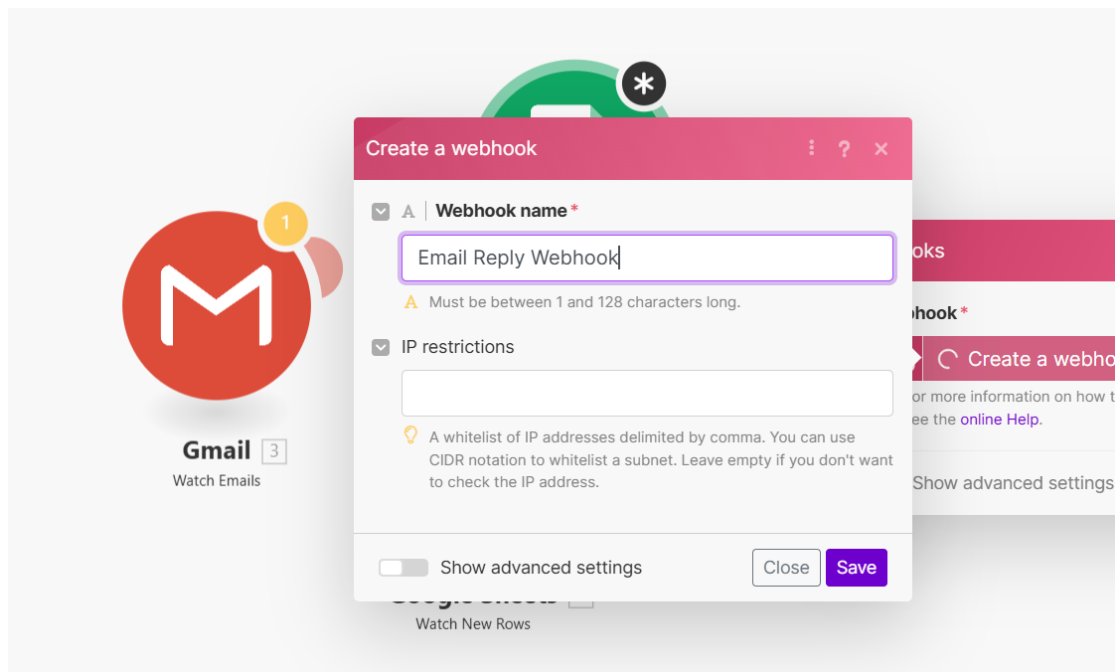
2.3 Google Sheets – Status Update

Module: Google Sheets "Update a Row"
Configuration Details:

- **Row Identification:** Use the unique row identifier or row number from the "Watch Rows" module.

- **Fields to Update:**

  - **status:** Set to "Sent" if the email is sent successfully, or "Failed" in the case of an error.

  - **sent time:** Update with the current timestamp when the email is sent.

- **Data Mapping:** Ensure that each field is correctly mapped to its corresponding column in the Google Sheet.

2.4 Reply Monitoring and Notification


- **Webhook Setup:**

  - Create a webhook URL in Make.

  - Configure email service to forward reply emails to this webhook.

- **Processing Data:**

  - Upon receiving data via the webhook, update the reply status in the Google Sheet and trigger notification modules.

Notification Module :

- **Notification Channel:** This can be an email notification, Slack message, or any other communication channel.

- **Notification Content:** Include details like the sender's email, a snippet of the reply, and the corresponding lead's information.

---

3. Error Handling Logic
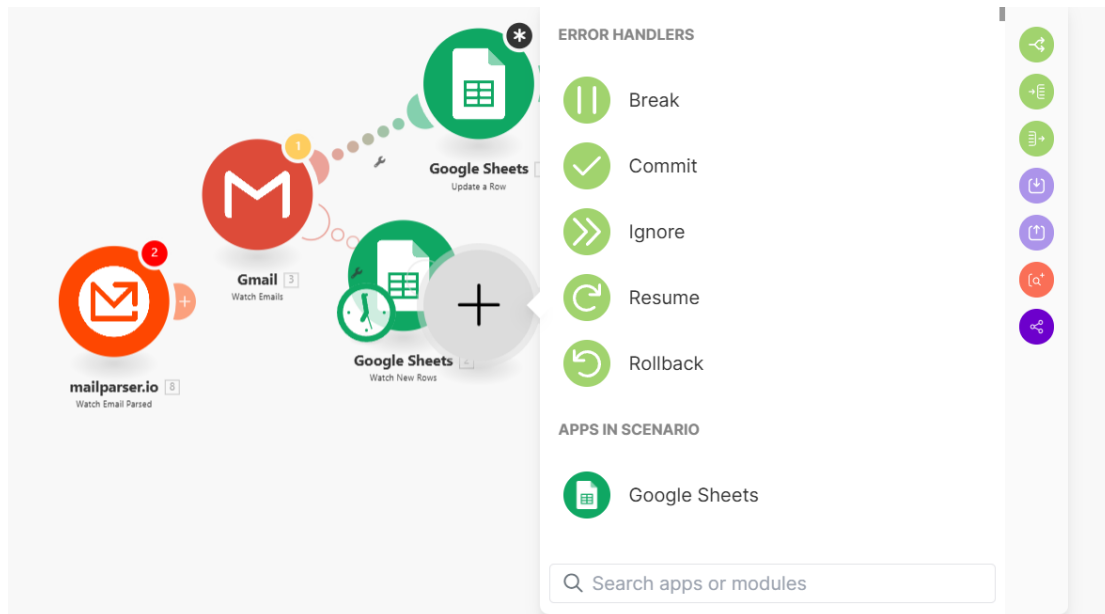
- **Example 1: Error in Google Sheets - Update a Row**

  - If an error occurs while updating a row in Google Sheets (e.g. network issue, incorrect data format, sheet deleted, etc.), I will do the following:

    - **Rollback:** If Google Sheets module supports transactions, Rollback would try to undo any partial changes made to the sheet during that update.

    - **Resume:** For instance, log the error to another sheet or send a notification. And then pass no data along, effectively skipping the update for that particular email, or use default values instead of the ones that caused an error.

    - **Break:** stop the scenario completely by using Break.

- **Example 2: Error in Gmail - Watch Emails**

  - If there is an error fetching emails from Gmail, maybe I just use Commit or Rollback as there is likely nothing to commit or roll back.

  - Using Ignore might be okay if missing an occasional email is acceptable, but it is generally not recommended as I could miss important data.

  - Resume would allow me to log the error or send a notification before continuing with the scenario.