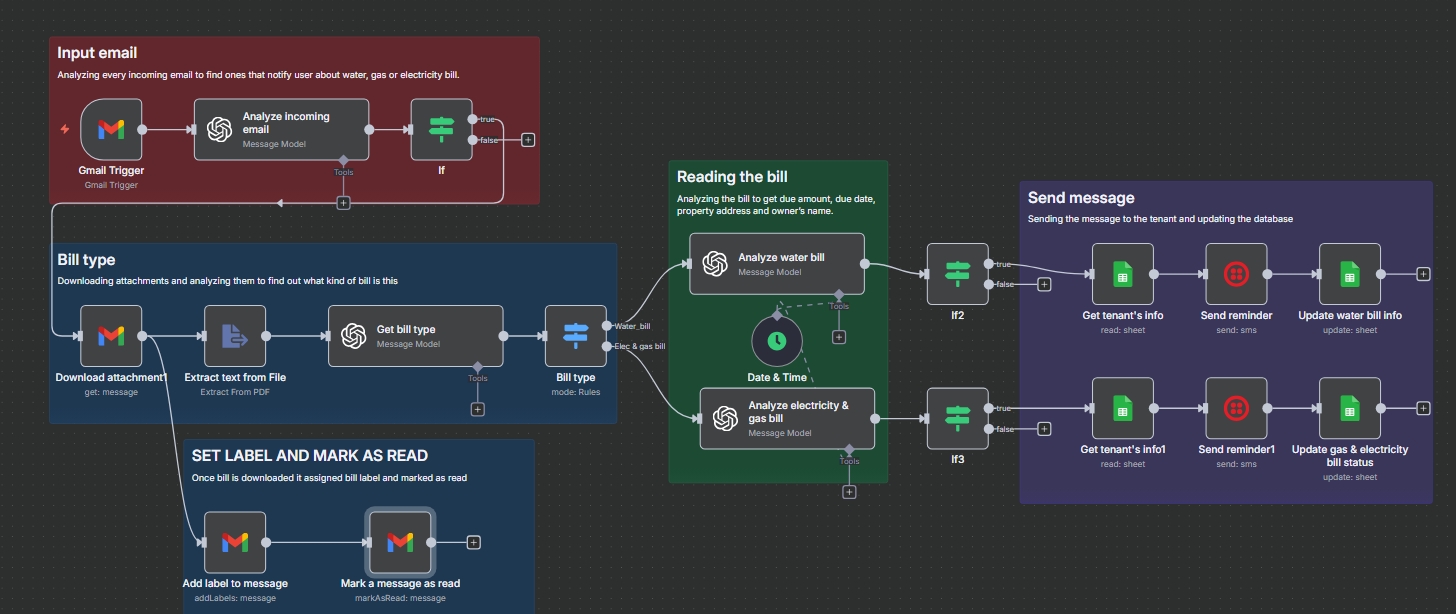
**Real-estate Agent**

### **Workflow: Automated Utility Bill Processing and Tenant Reminders**

**Objective:** To automatically parse incoming utility bill emails (water, electricity), extract key information, send an SMS reminder to the correct tenant, and update google sheets.

**Roles and Responsibilities:**

* **Automation System (like n8n):** The primary worker that executes all the steps.
* **Tenant:** The individual who receives the SMS reminder and is responsible for payment.
* **Property Owner:** The person who receives confirmation that the bill has been processed and the tenant has been notified.



Workflow

### **1. Workflow Overview**

* **1.1. Purpose:** The primary goal of this workflow is to completely automate the manual process of checking emails for utility bills, reminding tenants about payment due dates, and keeping a record for the property owner.
* **1.2. Scope:**
  + **In-Scope:** Monitoring a specific Gmail account, identifying water and electricity bills, extracting data from PDF attachments, looking up tenant info, sending SMS reminders, updating a Google Sheet log, and archiving the source email.
  + **Out-of-Scope:** Bill payment processing, handling tenant replies, and managing bills for utilities other than water and electricity.
* **1.3. Key Benefits:**
  + **Efficiency:** Saves hours of manual work each month.
  + **Accuracy:** Reduces human error in data entry and communication.
  + **Timeliness:** Ensures tenants receive timely reminders, preventing late fees.
  + **Record Keeping:** Creates a perfect, auditable log of all processed bills.

### **2. Prerequisites & Setup**

Before this workflow can operate, the following systems must be set up and configured.

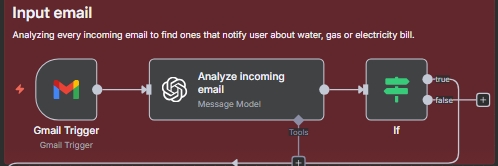
* **n8n Account:** A running instance of n8n (cloud or self-hosted) where the workflow is built.
* **Gmail Account:** A dedicated email account where utility bills will be sent. Credentials must be created in n8n to allow access.
* **SMS Gateway Service (e.g., Twilio):** An account with an SMS provider to send the text message reminders. API credentials must be added to n8n.
* **Google Sheets:** A Google account with Sheets API. Credentials must be added in n8n
* **Columns in Sheets: Property address, Tenant Name, Tenant Phone, Tenant Email, Property owner, Amount Water Bill, Water bill Account Number, Water Bill Billing Date, Water Bill Text Status, Last processed for Water Bill, Amount AEP Electric Bill, AEP Account Number, AEP Billing Date, AEP Bill text Status, Last processed for AEP bill.**

### **3. Detailed Workflow Breakdown**

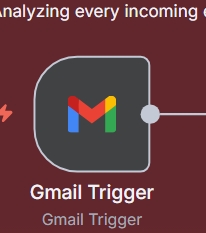
This workflow is divided into five distinct modules that execute in sequence.

#### **3.1. Module 1: Input Email (The Red Box)**

This module is the trigger. It listens for new emails and decides if they are relevant.

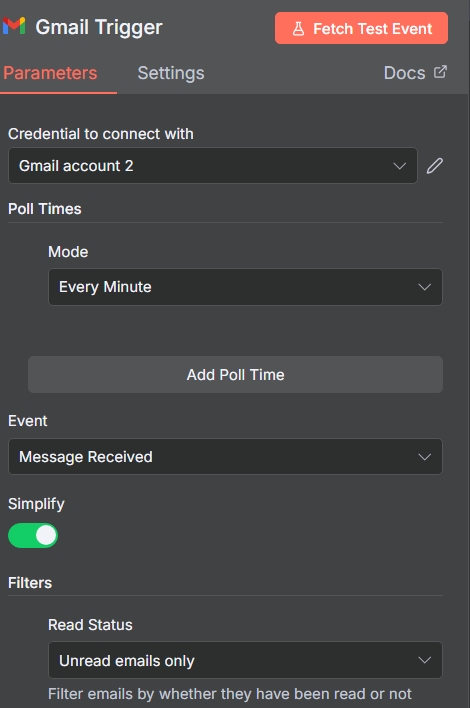
1. **Gmail Trigger:** The workflow starts the moment a new email arrives in the connected Gmail account.
2. **Analyze Incoming Email (IF Node):** This is a filter. It immediately checks the email's **subject line or sender** to see if it's likely a bill (e.g., subject contains "Your Water Bill"). 

**Each node overview**



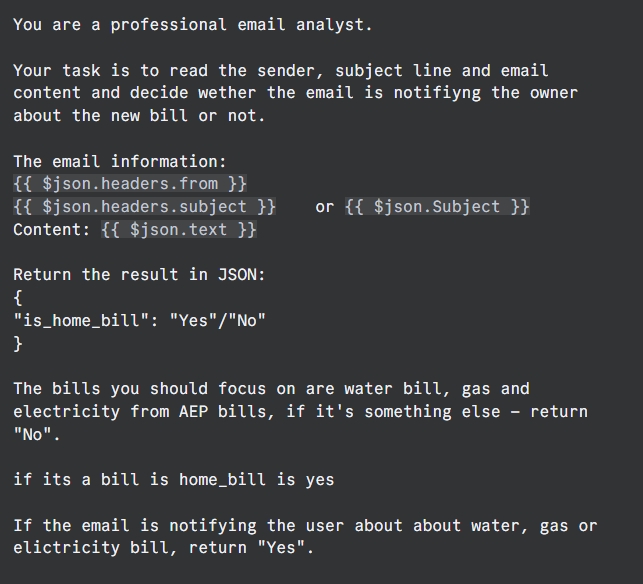
## **GMAIL Trigger**

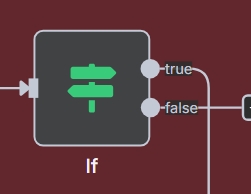
* **Purpose:** This node starts the workflow. It constantly monitors your Gmail inbox and automatically begins the process the instant a new email arrives.
* **Configuration:** It is linked to your Gmail account using credentials. Its primary setting is to trigger on "new email."
* **Output:** It passes the full data of the new email, including the sender, subject, body, and a unique **Message ID** for the attachments, to the next node.

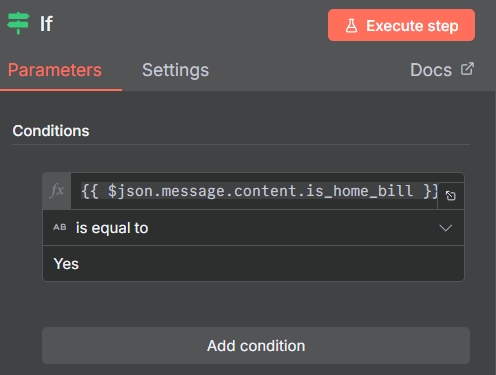


Gmail trigger config

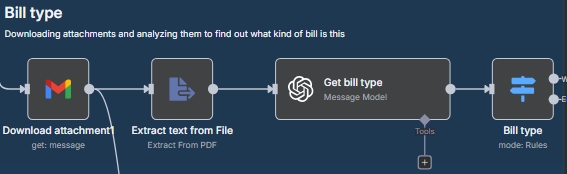
**Analyze incoming email (AI Message Model):**

* **Purpose:** This node acts as a smart filter. Instead of just looking for simple keywords, it uses AI to understand the *intent* of the email.
* **How it Works:** It takes the subject and body text from the email and asks an AI model a question like: "Based on the following text, is this email about a water, gas, or electricity bill? Please respond with only 'yes' or 'no'."
* **Prompt : **

**IF Node ("Analyze incoming email")**

* **Purpose:** This is a crucial gatekeeper. It checks if the incoming email is actually a bill worth processing, preventing the workflow from running on personal emails or spam.
* **Configuration:** You set conditions to check the email's data from ai agent output is yes or no
* **Output:** If the conditions are true, it allows the email's data to pass through. If false, the workflow stops immediately.

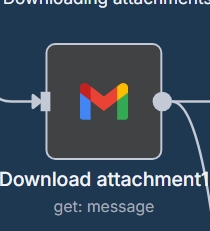
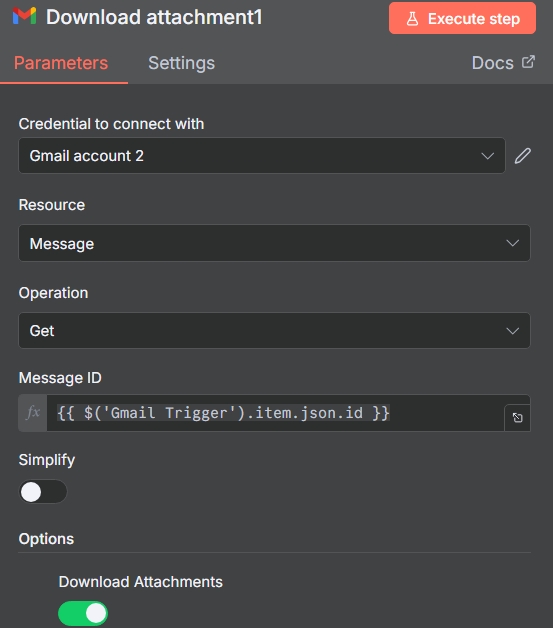
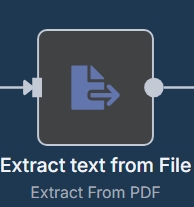
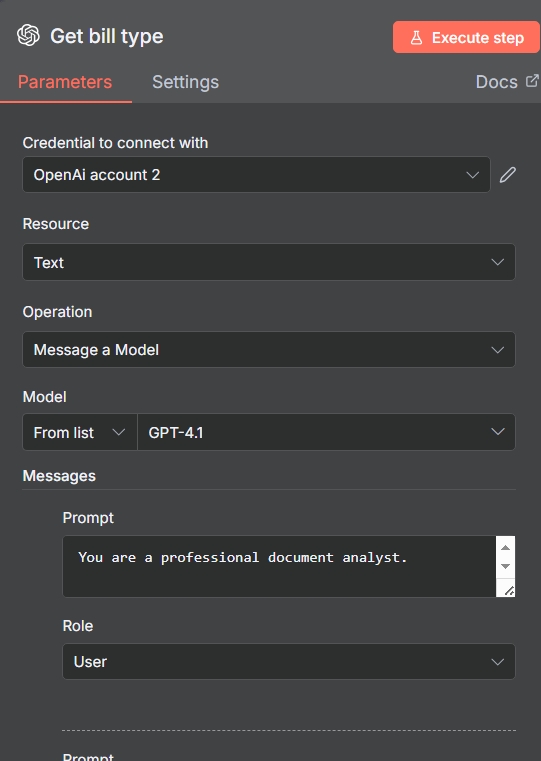
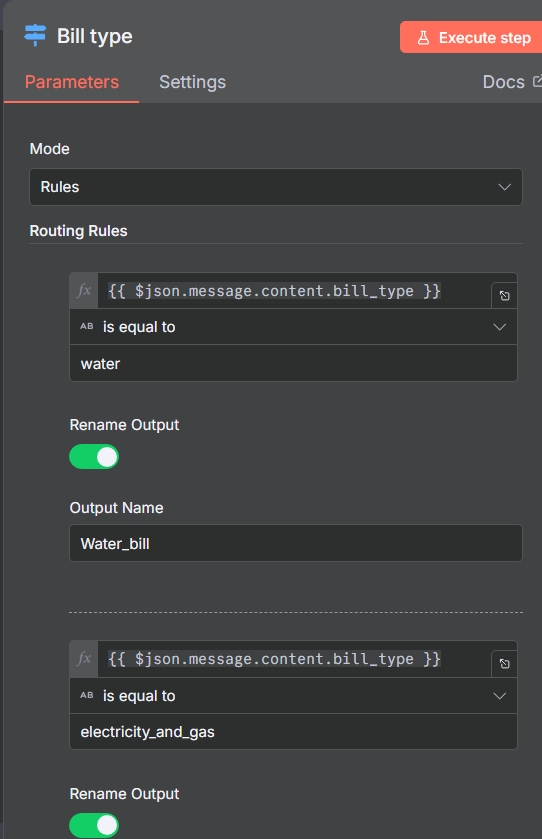
#### **3.2. Module 2: Bill Type (The Blue Box)**

This module downloads the attached bill and identifies what kind it is.

1. **Download Attachment:** Gets the PDF file attached to the email.
2. **Extract Text from File:** Reads all the text content from the PDF document.
3. **Get Bill Type (AI Model):** The extracted text is sent to an AI model with a simple prompt.
4. **Prompt Example:** "Read the following text from a bill and determine if it is a 'Water' bill, 'Electricity' bill, or 'Other'. Respond with only one of those three words. Text: [Full text from PDF]"
5. **Bill Type (Switch Node):** This node directs the workflow down a different path based on the AI's response ('Water' or 'Electricity'). If the AI returns 'Other', the workflow stops.

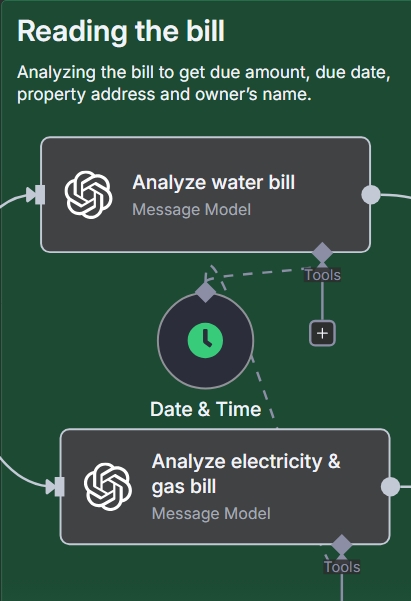
**Each node overview**

If the email is a bill, this module figures out exactly what kind of bill it is.

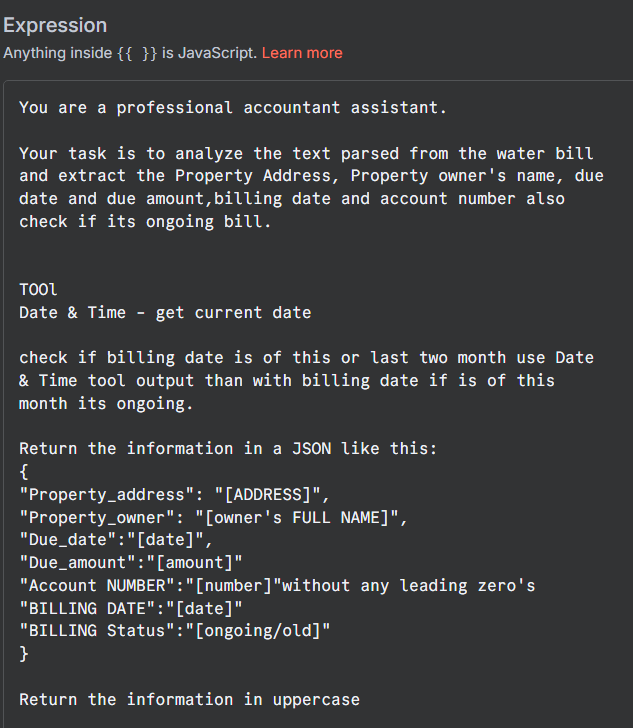
* **1. Download attachment (Gmail Node):**
  + **Purpose:** To get the actual bill file, which is usually a PDF.
  + **How it Works:** It uses the unique ID of the email to find and download the attached file,
* **2. Extract text from File:**
  + **Purpose:** To read the contents of the PDF file.
  + **How it Works:** It takes the PDF data and extracts all the text from it, turning the visual document into plain text.
* **3. Get bill type (AI Message Model):**
  + **Purpose:** To classify the bill.
  + **PROMPT:**
  + **How it Works:** It sends the full text of the bill to an AI with a very specific question: "Read the text. Is this a 'water' bill or an 'electricity' bill? Respond with only one of those words."
* **4. Bill type (Switch Node):**
  + **Purpose:** To direct the workflow down the correct path.
  + **How it Works:** 
  + If it receives the word water bill from the AI, it sends the data down the top path. If it receives electricity, it sends it lower path.

#### 

#### **3.3. Module 3: Reading the Bill (The Green Box)**

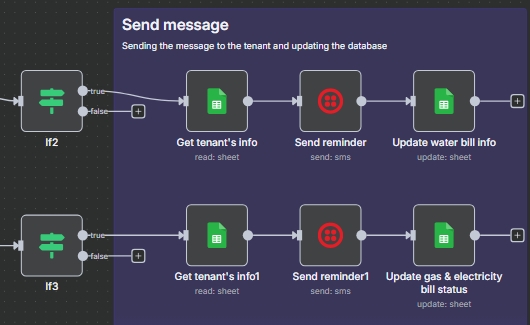


* **Analyze Water/Electricity Bill (AI Model):** The full text from the PDF is sent to ai agent prompt(example)



* **Date & Time Node**
  + **Purpose:** To get the current date for logging purposes, creating a timestamp of when the automation ran.
  + **Configuration:** It is configured to output the current date in a specific format, like YYYY-MM-DD.
  + **Output:** It outputs a string containing today's date (e.g., 2025-08-04).

**3.4. Module 4: Send Message (The Purple Box)**

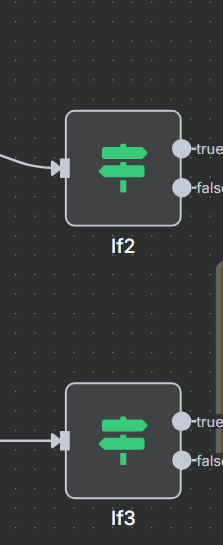


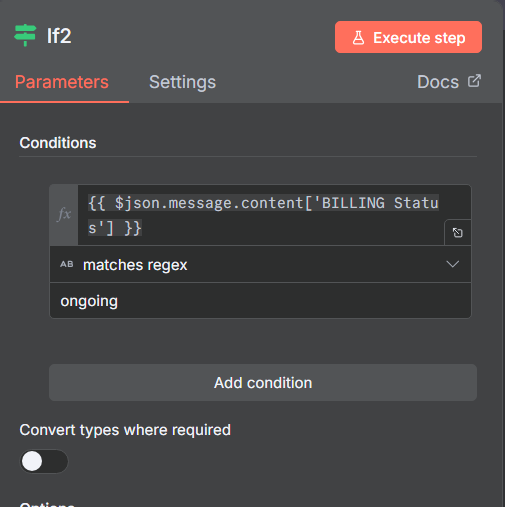
This is the action module. It communicates with the tenant and updates the records.

1. **Get Tenant's Info:** The workflow takes the address from the previous AI step and looks it up in the **Tenants Info Sheet** to find the matching **Tenant Name** and **Tenant Phone**.
2. **Send Reminder (SMS Node):** The system uses the SMS Gateway to send a message.
   * **Message Template:** "Hello [Tenant Name]. Your [Bill Type] bill for $[Amount] is due on [Due Date]. Thank you."
   * All the bracketed fields are filled in automatically with data from the previous steps.
3. **Update Bill Info (Google Sheets Node):** The workflow adds a new row to the **Bill Log Sheet**, filling in each column with the data it has gathered (Bill Type, Amount, Due Date, etc.) and sets the **Status** to "Reminder Sent ".

**Each node overview**

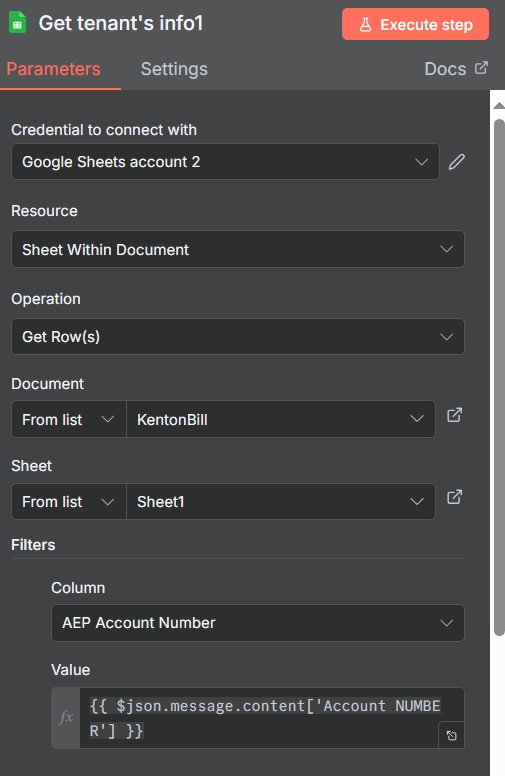
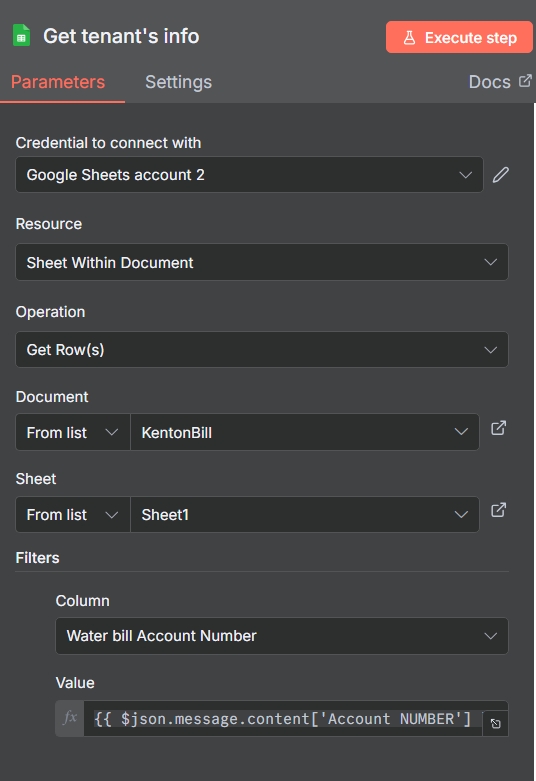
* **If2 / If3 Nodes**
* **Purpose:** To act as a final safety check, ensuring that that give bill is new
* **Configuration:** It is configured with a condition to check if the data from the previous AI node is valid (e.g., Billing status is ongoing).
* **Output:** If the condition is true, it passes the data along. If false, it stops that path of the workflow to prevent errors.





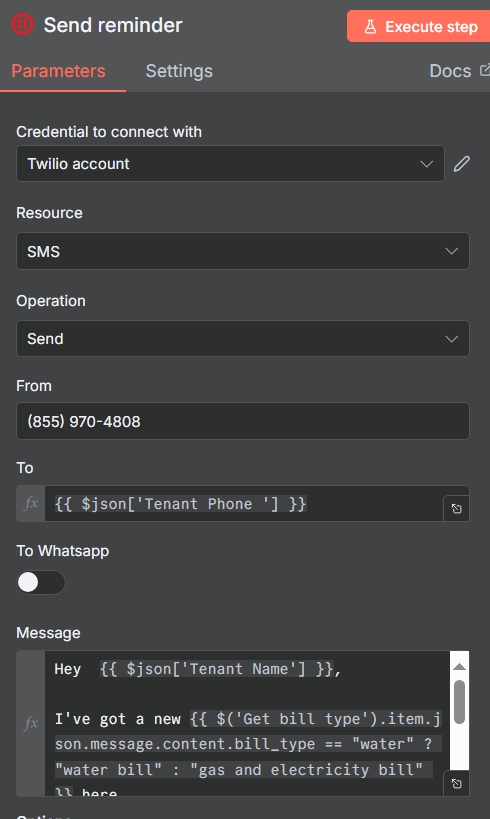
**2. Get tenant's info (Google Sheets Node)**

* **Purpose:** To find the correct tenant's name and phone number from your master database.
* **Configuration:** It is configured to "read: sheet" using the "Find Rows" operation. It searches the water and electricity account number column using the address provided by the AI node.

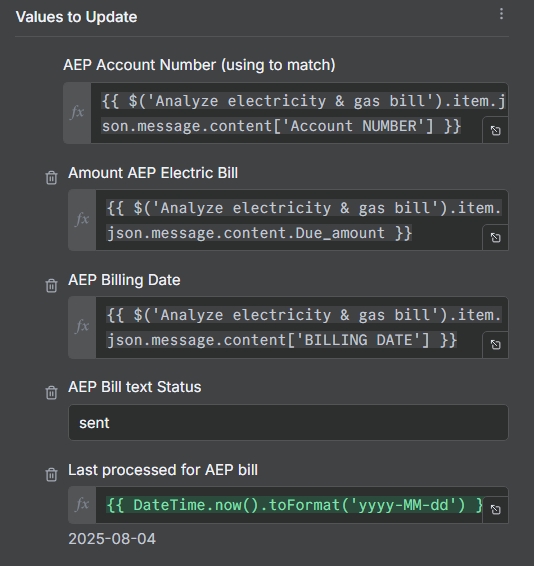
Get tenant info by matching account number from electricity or water bill

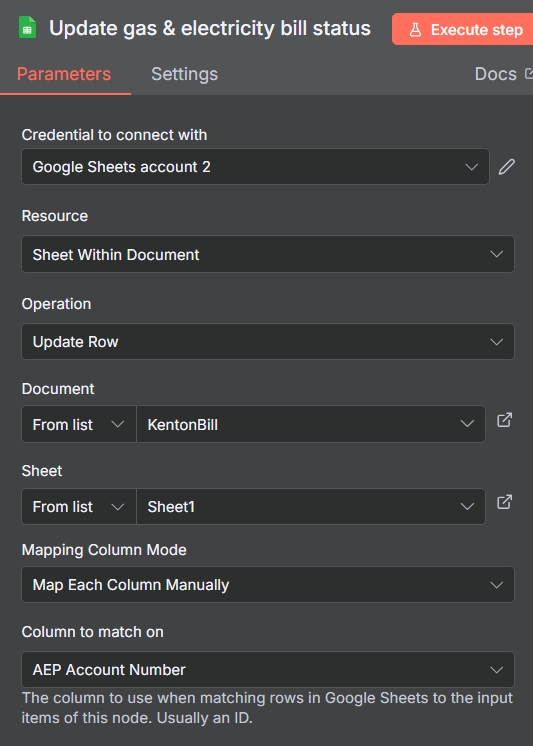
**3. Send reminder (SMS Node)**

* **Purpose:** To automatically send a personalized payment reminder to the tenant.
* **Configuration:** It is configured with your SMS provider's credentials. The message body is dynamically built using data from previous steps (e.g., "Hello {{Tenant Name}}, your bill of ₹{{amount}} is due.").
* **Output:** It outputs a status ID and confirmation that the SMS was sent successfully.

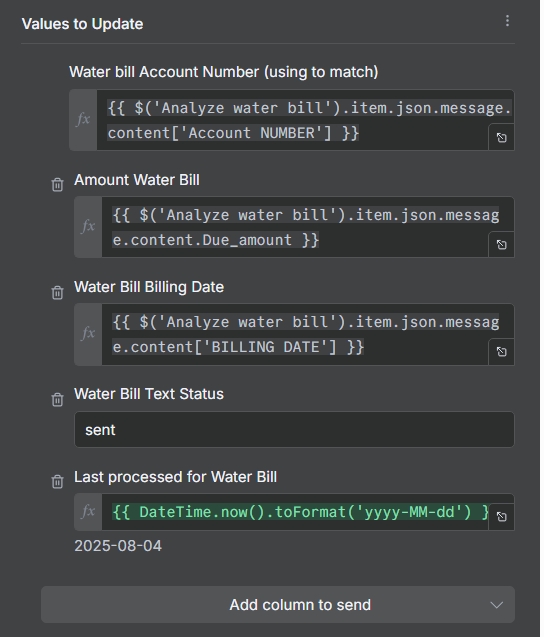
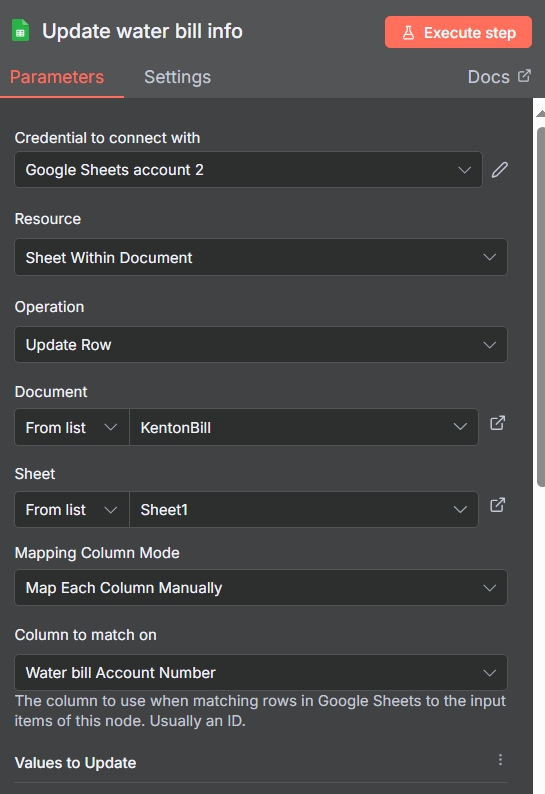
Sms content and twilio node config

**4. Update... bill status (Google Sheets Node)**

* **Purpose:** To create a permanent record of the action, confirming that the bill has been processed.
* **Configuration:** It is configured to "update: sheet". It uses the Row Number from the "Get tenant's info" step to update the correct row. It then fills specific columns like Amount Water Bill and Last processed for Water Bill with the new data.
* **Output:** It outputs a confirmation that the specified row and cells were successfully updated.



Electricity bill update in sheets



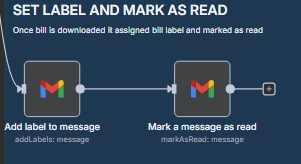
Water bill info update in sheets

#### 

#### 

#### 

#### **3.5. Module 5: Set Label and Mark as Read (Cleanup)**



The final module archives the email to keep the inbox clean.

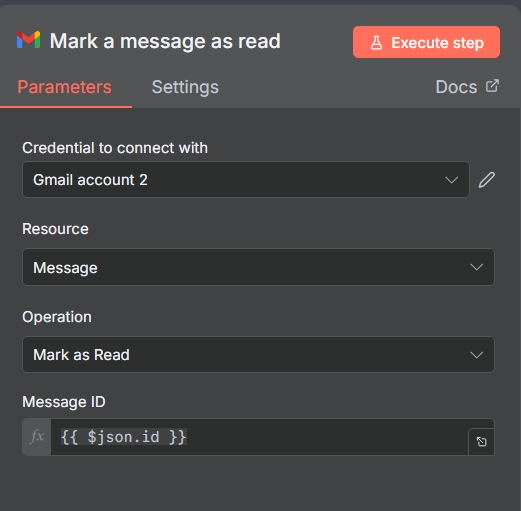
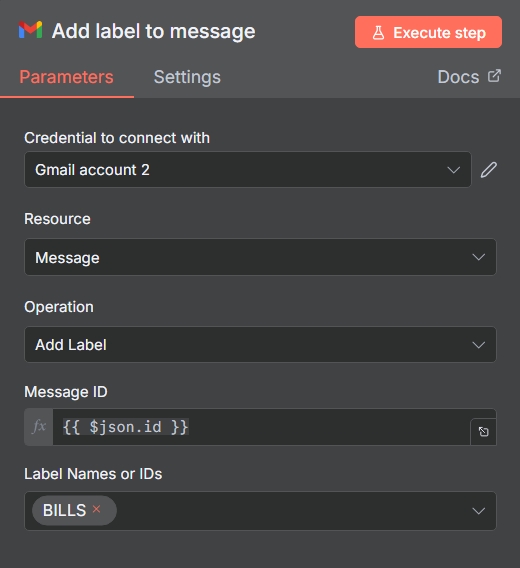
1. **Add Label to Message:** Applies a Gmail label like "Processed" to the original email.
2. **Mark a Message as Read:** Changes the email's status from unread to read.

**1. Add label to message (Gmail Node)**

* **Purpose:** To organize the Gmail inbox by visually marking the email as processed.
* **Configuration:** It is configured to use the Message ID from the very first trigger node and apply a pre-defined label (e.g., "Processed").
* **Output:** It outputs a success message.

**2. Mark a message as read (Gmail Node)**

* **Purpose:** To clean the inbox by moving the processed email out of the unread list.
* **Configuration:** It is configured to use the same Message ID from the trigger node and change the email's status from unread to read.
* **Output:** It outputs a success message.



GMAIL BILLS LABEL and READ MARKING CONFIG