**Lead Mailer AI Automation - Learning Guide**

**What This Agent Does**

Manual data entry like logging daily weather forecasts or temperatures for specific locations is repetitive and time-consuming. In many operational, planning, or educational use cases, teams want a structured weather history or forecast to align with business needs (e.g., planning field activities, adjusting schedules, or documenting environmental conditions).

**Why This Is Useful**

If you're new to automation or AI agents, this workflow shows you:

* Fetching weather data from an external weather API (Open-Meteo)
* Formatting relevant information (e.g., temperature, date)
* Logging it directly into a structured Airtable base

This removes the need for repetitive manual input and ensures that data is consistent and accessible in real-time.

**Key Concepts Covered**

* **Form Triggering**: Capturing user inputs directly into the workflow
* **Switch Logic**: A basic introduction to branching workflows
* **Email Automation**: Sending custom responses and routing messages
* **Labelling and Organization**: Automating inbox management

**How the Agent Works (Step-by-Step)**

**Node 1: Form Node (Trigger)**

* **Purpose**: Starts the automation when someone fills the webform manually
* **Setup**: Configured as a POST endpoint, ready to accept input or just act as a timer-based trigger.

**Node 2: Set (City Name & Coordinates)**

* **Purpose**: Predefines the required inputs — in this case, the location details (e.g., latitude and longitude).
* **Special Logic**: This can be expanded to support dynamic inputs from the webhook or loop through multiple locations in the future.

**Node 3: HTTP Request – Open-Meteo API**

* **Purpose**: Fetches real-time or forecast weather data based on the input coordinates.
* **Custom Config**: Uses the “daily” parameter to extract max/min temperatures.
* **API Reference**: <https://open-meteo.com>

**Node 4: Code Node (Extract Required Fields)**

* **Purpose**: Extracts relevant values like: Date, Maximum temperature, Minimum temperature
* **Logic**: Loops through the returned JSON array and maps only the necessary values for Airtable.

**Node 5: Airtable – Create Records**

* **Purpose**: Logs the extracted weather data into a predefined Airtable base and table.
* **Fields mapped**: Date, Max Temp, Min Temp
* **Error Handling**: Can be extended to handle duplicates or missing fields.

**Tools Used and Why**

* **n8n**: Open-source workflow automation tool used to orchestrate the entire flow
* **Form Trigger in n8n**: Provides a clean user-facing form
* **Switch Node in n8n**: Handles decision-making logic
* **Gmail Nodes**: Help send and organize emails efficiently

You can use alternate tools like Make or Zapier to achieve the same result.

**Build It Yourself – Quick Walkthrough**

Before you run this agent, you need to complete a few setup steps:

**1. Airtable Setup**

* Go to: <https://airtable.com/create/tokens/new>
* Generate a **personal access token**
* Grant access to the **base** and **table** you want to log weather data into
* Note down:
  + **Base ID**
  + **Table name**
  + **Field names** (ensure they match the ones used in the workflow)

**2. Open-Meteo API**

* Visit: <https://open-meteo.com>
* Identify the API endpoint you’ll use (e.g., /forecast)
* Parameters to include:
  + Latitude / Longitude
  + Daily fields (e.g., temperature\_2m\_max, temperature\_2m\_min)
  + Timezone

**3. n8n Webhook Setup (Optional)**

* If using a webhook to manually trigger the flow:
  + Enable webhook in **production** mode (not just test)
  + Copy the generated webhook URL to use as your trigger source

**Suggested Extensions**

* Adding multiple cities using a loop
* Sending daily summary emails
* Integrating with Slack or MS Teams for weather alerts
* Using cron to schedule it daily