

# n8n Workflow - Personal Weekly Meal Plan Generator

## What it does:

Generate weekly meal plan with recipes and shopping lists for the meal plan based on user inputs

## Why choose to build it:

There are people are struggling with what and how to eat to achieve their personal goals, and if they do research every week, that will cost a lot of time and efforts

## Why automating this process is logical and useful:

1. Manually create personal meal plan is time consuming, this workflow saves time
2. Making meal plan is repetitive but with different contents, ai agent could do exactly the same thing
3. Provide a clear document to track food

## Who benefits from it:

1. Anyone are busy with lives, have no time to think about what to eat and buy, but want to eat healthy and save time and money for their meal preparation
2. Fitness coaches
3. Meal-prep companies
4. Influencers & content creators

## Where it would be applied in real life:

1. Health & fitness coaching programs in any platforms
2. Family planning
3. Social media

## Step 1 Input User Data

- **Google Sheet Node:** A sheet links to a meal plan preferences Google Form. Stored user data.

**User Input Data**

**Parameters**

Credential to connect with

n8n Meal Plan Generator

**Poll Times**

Mode

Every Minute

Add Poll Time

**Document**

From list: Meal Preferences & Nutrition ...

**Sheet**

From list: Form Responses 1

**Trigger On**

Row Added

**Options**

Value Render

Formatted

Add option

**Fetch Test Event**

### 1. Options:

- Value Render: Formatted
  - To structure the Timestamp output
- Sort: Sort all the users input from the Google Sheet node and order them with TimeStamp (the time updated) in descending order

↓↑ Sort User Data Based o...

Execute step

Parameters Settings Docs ↗

Type : Fixed Expression

Simple

Fields To Sort By

Field Name: Timestamp

Enter the field name as text

Order: Descending

Add Field To Sort By

Options

No properties

Add Field

3 / 10

1. Field Name: Timestamp
  - Order: Descending fields
2. Order: Descending
  - The newest item will be listed first on the top to be chosen

- **Limit:** Limit to get only 1 (the newest one) data row from Google sheet to process

Limit User Data

Parameters    Settings    Docs

Max Items: 1

Keep: First Items

Execute step

1. Max Items: 1
  - Only past 1 newest item (avoid repetition)
2. Keep: First Items
  - Get the newest records
  - **Week Tag:** Codes to help to generate week periods for the plan, which are convenient for users to catch up
1. Generate updated week period for documents that with current data

---

## Step 2 Generate Meal Plan

- **AI Agent:** AI agent with Google Gemini chat model to generate weekly meal plan details

**AI Agent Meal Plan**

**Execute step**

**Parameters**    **Settings**    **Docs ↗**

Tip: Get a feel for agents with our quick [tutorial](#) or see an [example](#) of how this node works

Source for Prompt (User Message) : Fixed Expression

Define below

Prompt (User Message)    Parameter: "pron"

```
=Create a simple 7-day meal plan (break  
fast, lunch, dinner, snacks) honoring:  
- Name: {{ $json['Your Name ' ] }}  
- Career/Lifestyle: {{ $json['Your Care  
er ' ] }}  
- Diet Type: {{ $json['What Type of Die
```

Require Specific Output Format

Enable Fallback Model

**Options**

System Message

You are a nutrition planning assistant.

Chat Model \*    Memory    Tool

The screenshot shows the 'Chat Model' section of the n8n interface. At the top, there's a 'Parameters' tab (which is active), a 'Settings' tab, and a 'Docs' link. Below this, a dropdown menu is set to 'Google Gemini(PaLM) Api account'. There are also dropdown menus for 'Model' (set to 'models/gemini-2.5-flash') and 'Sampling Temperature' (set to '0.8'). A button labeled 'Add Option' is visible.

## 1. AI Model: Google Gemini

- Add options
- Sampling Temperature: 0.8 (the larger the number, the higher randomness ----> to help ai agent to generate different contents each execution although it may have the same user inputs)

## 2. Prompt:

- Create 7-day meal plan based on user inputs
- Set it generate different meal plan for each execution (even though it has the same user inputs) (not duplicated)

## 3. System Message:

- Extra information to structure output, such as not changing the week tag input, let ai agent act as nutrition planning assistant, and check the java codes to avoid errors
- **Code Format Meal Plan:** To structured the contents from AI agent to make it look better in the document

---

## Step 3 Generate Recipes

- **AI Agent:** AI agent with Google Gemini chat model to generate recipes and shopping list

## AI Agent Recipes

Execute step

Parameters    Settings    Docs

Tip: Get a feel for agents with our quick [tutorial](#) or see an [example](#) of how this node works

Source for Prompt (User Message)

Define below

Prompt (User Message)

You are a registered-dietitian-level recipe generator. Create simple, budget-conscious recipes for each meal in a 7-day plan. Respect allergies, restrictions, diet type, disliked foods, time

Require Specific Output Format

Enable Fallback Model

Options

System Message

You must output STRICT, valid JSON only.

Chat Model \*    Memory    Tool

 + +

The screenshot shows the n8n AI Model configuration interface. At the top, there's a header with the title "Chat Model1" and three tabs: "Parameters" (which is red and underlined), "Settings", and "Docs". Below the header, there's a section titled "Credential to connect with" with a dropdown menu set to "Google Gemini(PaLM) Api account". Under "Model", there's a dropdown menu set to "models/gemini-2.5-flash". In the "Options" section, there's a "Sampling Temperature" input field containing "0.8" and a button labeled "Add Option".

### 1. AI Model: Google Gemini

- Add options
- Sampling Temperature: 0.8 (the larger the number, the higher randomness ----> to help ai agent to generate different contents each execution although it may have the same user inputs)

### 2. Prompt:

- Create recipes and shopping lists according to the meal plan
- Set it generate different ingredients each time

### 3. System Message:

- Instruct ai gent to output strict java codes to lower possibility of errors
- **Code Format Recipes:** To structured the contents from AI agent to make it look better in the document

## Step 4 Send the Meal Plan

- **Copy file (Google Drive):** Since we have the meal plan template in the google drive, we use this node to to copy the template to pass it to the next step to fill in user information

- **Update a document (Google Doc):** Update user input to template

The screenshot shows the configuration interface for the 'Update a document' step in n8n. At the top, there's a title bar with the step name and a red 'Execute step' button. Below it, there are three tabs: 'Parameters' (which is selected), 'Settings', and 'Docs'. The 'Parameters' tab contains fields for 'Doc ID or URL' (set to `{{ \$json.id }}`), 'Simplify' (set to 'Fixed Expression' and turned on), and an 'Actions' section. The 'Actions' section includes dropdowns for 'Object' (set to 'Text') and 'Action' (set to 'Find and Replace Text'), and fields for 'Old Text' (set to `{{Name}}`) and 'New Text' (set to `{{ \$('Limit User Data').item.json['Your Name'] }}`). There's also a 'Match Case' toggle switch.

#### 1. Action: Find and Replace Text

- Find the original text in the copy template and replace it with ai agent contents and user input info
- **Download file:** Convert and download the google doc meal plan file to pdf for user
- **Send a message (Gmail):** Send the finalized meal plan to user with message and subject

The screenshot shows the configuration for the 'Send a message' step in n8n. The top bar includes the step name 'Send a message' and an 'Execute step' button. Below the header, there are tabs for 'Parameters' (which is active), 'Settings', and 'Docs'. The 'Parameters' section contains fields for 'Credential to connect with' (set to 'n8n Meal Plan Gmail'), 'Resource' (set to 'Message'), 'Operation' (set to 'Send'), 'To' (containing a formula: `{{ \$(\$('Limit User Data').item.json['Your Email']) }}`), 'Subject' (containing a formula: `Weekly Meal Plan ({{ \$(\$('Codes Format Meal Plan').item.json.WekPeriod })}} - {{ \$(\$('Limit User Data').item.json['Your Name']) }}`), 'Email Type' (set to 'HTML'), and 'Message' (containing sample text: 'Hi, Your personalized 7-day meal plan is now ready!'). A vertical scroll bar is visible on the right side of the configuration area.

1. To (email):
  - User input email from previous step
2. Subject:
  - Include week tag codes: show week dates