

# WhatsApp Restaurant Agent – Project Documentation (n8n + Google Sheets)

---

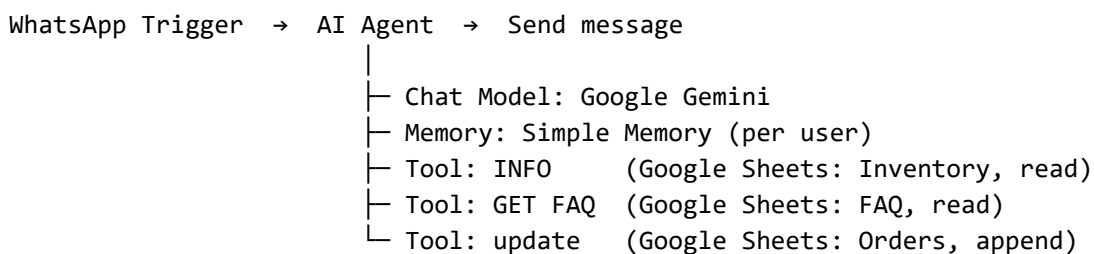
## Overview

This project implements a conversational WhatsApp assistant for a restaurant using n8n. The assistant is powered by an AI Agent (Google Gemini Chat Model), maintains short-term context with Simple Memory, and relies on Google Sheets as the source of truth for Inventory, FAQ, and Orders. Customers can ask FAQs, check menu/stock, and place orders. Confirmed orders are appended to the Orders sheet.

## Key Features

- WhatsApp conversation via Meta WhatsApp Business Cloud API
- AI Agent with tool-use (n8n Google Gemini Chat Model)
- Simple Memory for per-user session context
- Google Sheets storage:
  - Inventory (menu, quantity, status)
  - FAQ (question/answer)
  - Orders (appended on confirmation)
- Natural order flow: name → item → quantity → stock check → confirm → append order
- Robust outbound message mapping so the chat always returns a reply

## High-Level Architecture



## Google Sheets Schema

Create one spreadsheet with three tabs:

Inventory (exact columns):

- Food Item (text)
- Quantity (number)
- Status ("Available" or "Out of Stock")

FAQ (exact columns):

- Question (text)
- Answer (text)

Orders (exact columns):

- Customer Name (text)
- Food Item (text)
- Quantity Ordered (number)
- Order Date (yyyy-MM-dd)
- Status (e.g., "Confirmed")

Column names must match exactly; the tools rely on them.

## Prerequisites

- Meta Business Manager access and a WhatsApp Business Account with a phone number
- A System User (Admin) and a long-lived access token with scopes: whatsapp\_business\_management, whatsapp\_business\_messaging
- n8n (Cloud or self-hosted)
- Google account and a Sheet prepared with the tabs above
- Google OAuth client (or Service Account) set up as an n8n credential

## n8n Credentials

- WhatsApp account credential: paste the System User access token
- Google Sheets credential: OAuth2 (add the Redirect URI shown by n8n) or Service Account

## Workflow Nodes

- WhatsApp Trigger – receives inbound messages
- AI Agent – orchestrates model + memory + tools
- Google Gemini Chat Model – large language model
- Simple Memory – per-user context (session key = sender WA ID)
- INFO – Google Sheets (Inventory, read rows)
- GET FAQ – Google Sheets (FAQ, read rows)
- update – Google Sheets (Orders, append row)
- Send message – sends the AI Agent's reply to WhatsApp

## AI Agent – Settings

- Tool choice: Auto (or Always use tools / Required)

- Max tool calls per turn: 2

## AI Agent – System Message

You are a smart food ordering assistant for “MDA Restaurant”.

### TOOL POLICY

- Tools you can call:
  - 1) INFO → read 'Inventory' sheet.
  - 2) GET FAQ → read 'FAQ' sheet.
  - 3) update → append a new row to 'Orders' sheet.

### ORDER FLOW

- If the user wants to place an order, collect in this order:
  - (a) Customer name, (b) Food item, (c) Quantity (number).
- Then call INFO to check the item. Treat an item as available only if Status="Available" AND Quantity>0.
- If not available → apologize and list available options (from INFO). Do NOT call update.
- If available → confirm to the user, THEN call update and fill these keys: Customer\_Name, Food\_Item, Quantity\_Ordered, Status="Confirmed".

### GENERAL

- For hours/delivery/refund/how to order → call GET FAQ.
- After any tool call, reply with a short WhatsApp-style message.

## Simple Memory – Session Key (Expression)

```
{{ $('WhatsApp Trigger').item.json.messages?.[0]?.from }}
```

## INFO Tool (Google Sheets – Inventory, read)

Purpose: menu/availability (Inventory read).

Tool Description:

Use for menu/availability. Sheet: Inventory (Food Item, Quantity, Status).

Only treat entries with Status="Available" and Quantity>0 as in stock.

Filters:

- Status = Available (constant)
- Food Item = (enable the parameter so the Agent can pass the user's requested dish)

## GET FAQ Tool (Google Sheets – FAQ, read)

Purpose: general questions (FAQ read).

Tool Description:

Use for general questions (hours, delivery, refund, payments, how to order). Return the matching Q/A.

Filters:

- Question = (enable the parameter so the Agent can pass the user's text)

## update Tool (Google Sheets – Orders, append)

Use only after an order is confirmed in inventory.

Column mappings (Agent-provided where noted):

- Customer Name → \$fromAI('Customer\_Name')
- Food Item → \$fromAI('Food\_Item')
- Quantity Ordered → \$fromAI('Quantity\_Ordered')
- Status → \$fromAI('Status')
- Order Date → Expression: {{ \$now.format('yyyy-MM-dd') }}

## Send message – Expressions

Recipient (dynamic):

```
{{ $( 'WhatsApp Trigger' ).item.json.messages?.[0]?.from  
  ?? $( 'WhatsApp Trigger' ).item.json.contacts?.[0]?.wa_id }}
```

Text Body (robust mapping):

```
{{  
  $( 'AI Agent' ).item.json.output  
  ?? $( 'AI Agent' ).item.json.messages?.[0]?.content?.[0]?.text  
  ?? $( 'AI Agent' ).item.json.response  
  ?? $( 'AI Agent' ).item.json.result  
  ?? 'Sorry, I had trouble generating a reply. Please try again.'  
}}
```

## Usage Examples

- “Hi” → bot greets with options
- “Place an order” → bot collects name, item, quantity
- “Mutton Thali 2 for Sam” → bot checks Inventory, confirms, appends to Orders
- “What are your opening hours?” → bot reads FAQ and replies

## Troubleshooting

- No reply after first message → verify Text Body expression above.
- Authorization failed (Send node) → WhatsApp token expired. Create a System User token and update the credential.
- INFO returns empty → column names/values mismatch (Status must be 'Available', exact case).
- Agent never uses tools → Tool choice = Auto/Required, clear tool descriptions, Max tool calls = 2.

## Security

- Store tokens in n8n Credentials (never hardcode in workflow JSON).

- Restrict spreadsheet sharing (principle of least privilege).
- Rotate WhatsApp tokens regularly.

## Appendix – Expressions (Copy/Paste)

Session key:

```
{{ $('WhatsApp Trigger').item.json.messages?.[0]?.from }}
```

Recipient:

```
{{ $('WhatsApp Trigger').item.json.messages?.[0]?.from ?? $('WhatsApp Trigger').item.json.contacts?.[0]?.wa_id }}
```

Order Date:

```
{{ $now.format('yyyy-MM-dd') }}
```

Robust Text Body mapping:

```
{{  
  $('AI Agent').item.json.output  
  ?? $('AI Agent').item.json.messages?.[0]?.content?.[0]?.text  
  ?? $('AI Agent').item.json.response  
  ?? $('AI Agent').item.json.result  
  ?? 'Sorry, I had trouble generating a reply. Please try again.'  
}}
```

Generated on 2025-09-09 08:29:00 UTC