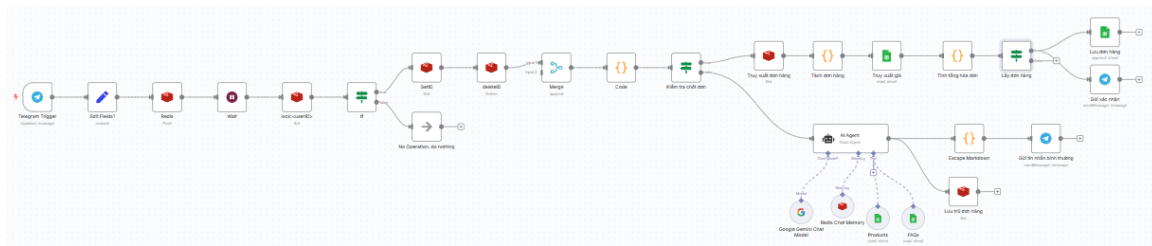


# TÀI LIỆU HƯỚNG DẪN TRIỂN KHAI CHATBOT CHĂM SÓC KHÁCH HÀNG CHO SHOP ÁO QUẦN TRÊN N8N

## I. Giới thiệu

Tài liệu này hướng dẫn chi tiết từng bước triển khai hệ thống chatbot chăm sóc khách hàng cho shop bán áo quần trên nền tảng n8n, tích hợp với Telegram, Google Sheets, Redis và mô hình AI (Gemini). Workflow gồm nhiều node, mỗi node thực hiện một chức năng cụ thể trong quá trình tương tác khách hàng: từ tiếp nhận tin nhắn, xử lý AI, đến lưu đơn hàng.

Workflow như sau:



Các node sử dụng trong workflow này gồm:

- Telegram Trigger
- Set (Edit Fields1)
- Redis (Push/Get/Set/Delete)
- Wait
- If
- Merge
- Code (nhiều mục đích: xử lý chuỗi, tính toán,...)
- AI Agent (LangChain)
- Google Gemini Chat Model
- Google Sheets Tool
- Telegram Send Message

## II. Mục tiêu hệ thống

- Tiếp nhận và xử lý tin nhắn từ người dùng qua Telegram.
- Truy vấn thông tin sản phẩm, hướng dẫn mua hàng.
- Hỗ trợ khách hàng đặt hàng, xác nhận đơn và lưu dữ liệu.
- Ứng dụng trí tuệ nhân tạo để trả lời linh hoạt theo ngữ cảnh.

### III. Chi tiết luồng xử lý và chức năng các node

#### *Bước 1: Nhận và xử lý tin nhắn Telegram*

- Telegram Trigger: Kích hoạt workflow khi có tin nhắn mới từ người dùng gửi đến bot Telegram.
- Edit Fields1 (Set): Trích xuất các thông tin: tele\_id (ID), tele\_msg (nội dung), tele\_user (tên người gửi).

#### *Bước 2: Ghi nhận và kiểm soát xử lý*

- Redis (Push): Lưu nội dung tin nhắn vào danh sách Redis theo ID người dùng.
- Wait: Tạm dừng xử lý để đảm bảo xử lý tuần tự.
- lock:<userID> (Redis Set): Tạo khóa tạm trong Redis để tránh trùng lặp xử lý trong 10 giây.
- If: Kiểm tra xem khóa đã tồn tại hay chưa để quyết định xử lý tiếp.

#### *Bước 3: Trích xuất lịch sử tin nhắn*

- GetID (Redis Get): Lấy lại toàn bộ lịch sử tin nhắn từ Redis.
- deleteID (Redis Delete): Xóa khóa sau khi lấy xong dữ liệu.
- Merge: Gộp thông tin người dùng và tin nhắn.
- Code: Ghép tất cả tin nhắn thành chuỗi để gửi AI xử lý.

#### *Bước 4: Xử lý tư vấn từ AI*

- AI Agent: Gọi AI Agent với prompt tư vấn. Giao tiếp thân thiện, dựa vào bảng ProductInfo và FAQs.
- Google Gemini Chat Model: Mô hình xử lý ngôn ngữ.
- Redis Chat Memory: Lưu nhớ phiên trò chuyện.
- Google Sheets Tool - Products: Truy vấn bảng sản phẩm được mua bán.
- Google Sheets Tool - FAQs: Truy vấn bảng câu hỏi thường gặp khi tư vấn vấn đề liên quan đến sản phẩm.

#### *Bước 5: Kiểm tra khi người dùng nhấn 'chốt đơn'*

- Kiểm tra chốt đơn (If): Nếu người dùng nhấn 'chốt đơn', bắt đầu xử lý đơn hàng.
- Truy xuất đơn hàng (Redis Get): Lấy lại thông tin đơn hàng đã lưu.

- Tách đơn hàng (Code): Trích xuất các trường từ nội dung đơn: tên, SĐT, sản phẩm, địa chỉ,...

#### *Bước 6: Tính hóa đơn và lưu đơn hàng*

- Truy xuất giá (Google Sheets): Lấy giá từ bảng sản phẩm theo tên và size.
- Tính tổng hóa đơn (Code): Tính tổng tiền: Giá  $\times$  số lượng + phí ship (nếu có).
- Lấy chốt đơn (If): Nếu đủ thông tin thì tiếp tục lưu đơn.
- Lưu đơn hàng (Google Sheets Append): Lưu thông tin đơn hàng vào bảng dữ liệu.
- Gửi xác nhận (Telegram Message): Gửi thông tin đơn hàng xác nhận cho khách.

#### *Bước 7: Trường hợp không phải đơn hàng*

- Lưu trữ đơn hàng (Redis Set): Lưu tạm nội dung đơn hàng xác nhận.
- Escape Markdown (Code): Làm sạch văn bản Markdown (chẳng hạn \*, \_) để tránh lỗi định dạng khi gửi Telegram..
- Gửi tin nhắn bình thường (Telegram): Gửi lại phản hồi của AI cho khách hàng.

## **IV. Cấu hình các node**

### *4.1 Telegram Trigger*

**Chức năng:** Kích hoạt workflow khi có tin nhắn từ người dùng Telegram gửi đến bot.

**Cấu hình:**

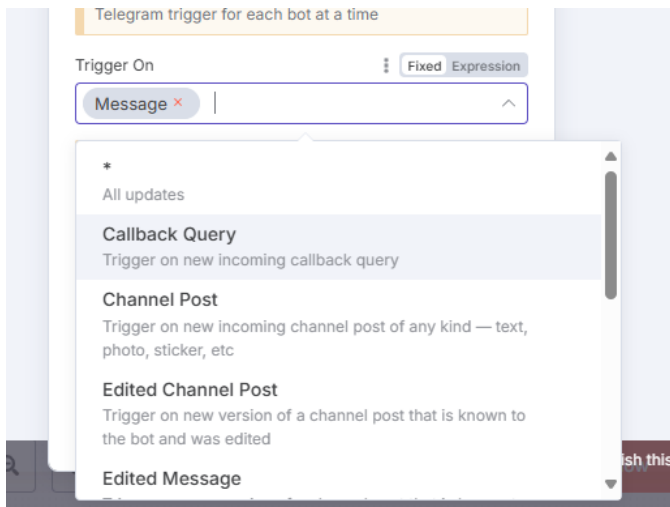
The screenshot shows the 'Telegram Trigger' configuration page. At the top, there's a 'Test step' button. Below it are tabs for 'Parameters', 'Settings', and 'Docs'. Under 'Parameters', there's a section for 'Webhook URLs'. A dropdown menu for 'Credential to connect with' is open, showing 'Telegram account' as the selected option. Below this, there's a '+ Create new credential' button. A 'Message' field is visible with a red 'x' icon. A yellow box contains a note: 'Every uploaded attachment, even if sent in a group, will trigger a separate event. You can identify that an attachment belongs to a certain group by media\_group\_id.' At the bottom, there's an 'Additional Fields' section with 'No properties' and an 'Add Field' button.

- Credential to connect with: chọn tài khoản Telegram bot đã cấu hình.

Nếu chưa có thì nhấn vào “Create new credential”

The screenshot shows the 'Telegram account 2' configuration page. At the top, there's a 'Save' button. Below it are tabs for 'Connection', 'Sharing', and 'Details'. Under 'Connection', there's a yellow box with the text 'Need help filling out these fields? Open docs'. Below this, there's an 'Ask Assistant' button for setup instructions. The 'Access Token' field is empty. The 'Base URL' field contains 'https://api.telegram.org'. At the bottom, there's a note: 'Enterprise plan users can pull in credentials from external vaults. More info'.

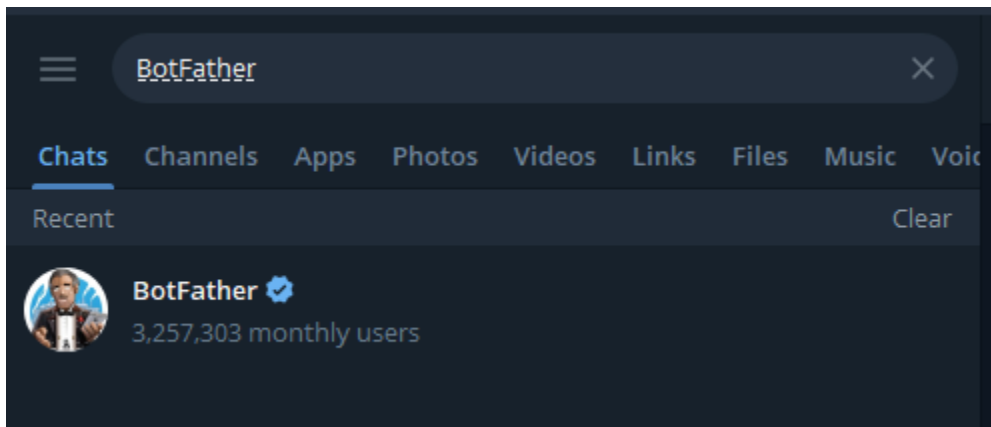
- Access Token: mã Token lấy từ BotFather trên Telegram (nhập ở phần cấu hình tài khoản Telegram).
- Base URL: giữ nguyên là `https://api.telegram.org`



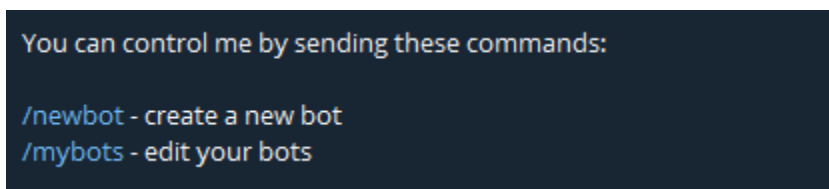
- Trigger On: chọn 'Message' để xử lý tin nhắn văn bản (có thể điều chỉnh thành 'Callback Query', 'Channel Post',... tùy mục đích sử dụng).

***\* Hướng dẫn tạo bot Telegram để sử dụng:***

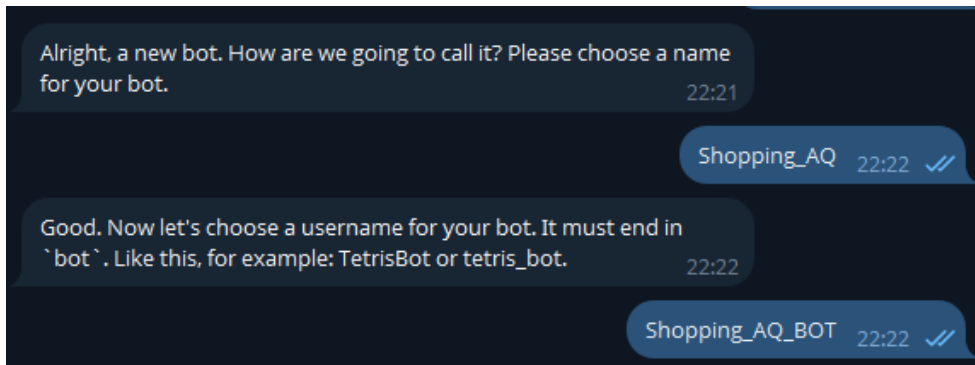
Bước 1: Tìm kiếm BotFather ở thanh tìm kiếm Telegram



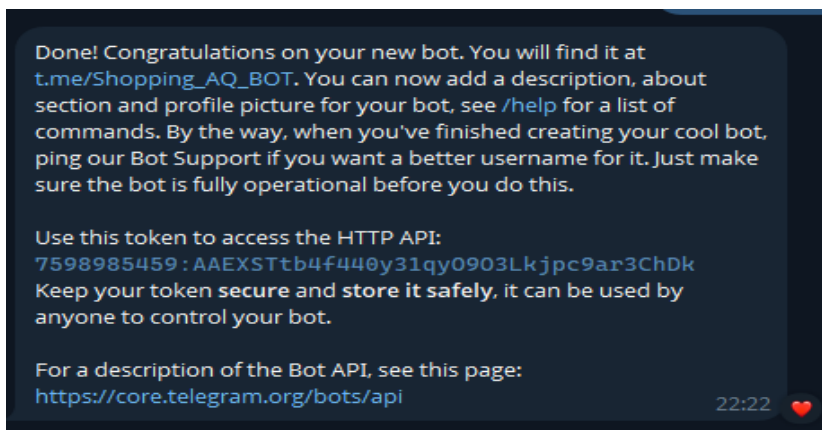
Bước 2: Để tạo 1 con bot mới nhấn “/newbot”



Bước 3: Đặt tên cho bot, lưu ý sau khi đặt tên cần nhắc lại 1 lần nữa với chữ BOT được đặt ở sau tên bot



Bước 4: Copy AccessToken để cấu hình tài khoản Telegram




#### 4.2 Edit Fields1 (Set)

**Chức năng:** Trích xuất các trường cần thiết từ tin nhắn Telegram để lưu trữ và xử lý về sau.

Cấu hình các biến:

**Cấu hình:**

 Edit Fields1

Test step

Parameters

Settings

Docs

Mode

Manual Mapping

Fields to Set

ID

A String

= {{ \$json.message.chat.id }}

messaging

A String

= {{ \$json.message.text }}

username

A String

= {{ \$json.message.from.first\_name || '' }}

- Mode:

Mode

Manual Mapping

**Manual Mapping**

Edit item fields one by one

**JSON**

Customize item output with JSON

Manual Mapping: thiết lập thủ công từng trường dữ liệu.

JSON: Xây dựng toàn bộ dữ liệu đầu ra ở dạng JSON một cách linh hoạt

-> Trong workflow chatbot, tôi chọn Manual Mapping để rõ ràng và dễ kiểm soát từng trường dữ liệu (trong node Edit Fields1). Nếu sau này bạn cần chuẩn bị một response phức tạp hoặc gửi dữ liệu đến API bên ngoài dưới dạng JSON → nên dùng JSON Mode.

- Fields to set:

Fields to Set

The screenshot shows the 'Fields to Set' configuration interface with three fields:

- ID**: Type 'A String', Expression: `{{ $json.message.chat.id }}`, Preview: 7223474955
- messaging**: Type 'A String', Expression: `{{ $json.message.text }}`, Preview: Cho đặt 2 cái áo đi (with a 'Fixed Expression' label)
- username**: Type 'A String', Expression: `{{ $json.message.from.first_name || '' }}`, Preview: Ally

ID: `{{ $json.message.chat.id }}` → mã người gửi.

- messaging: `{{ $json.message.text }}` → nội dung tin nhắn.

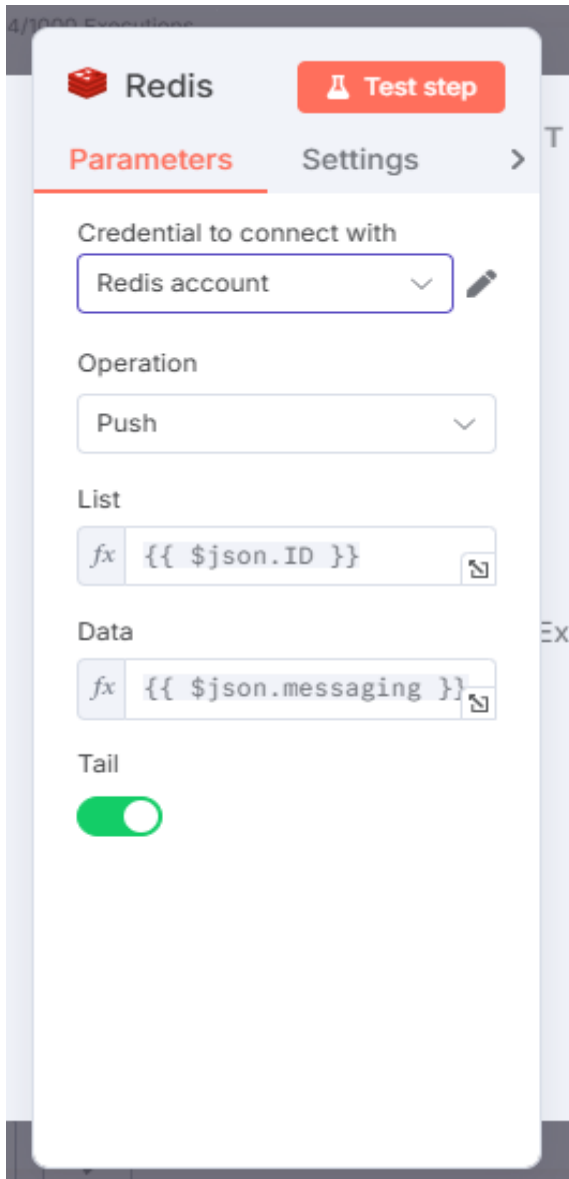
- username: `{{ $json.message.from.first_name || '' }}` → tên người gửi.

#### 4.2 Redis (Push)

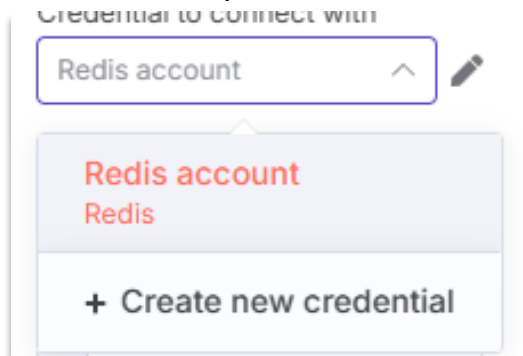
**Chức năng:** Node này lưu tin nhắn khách hàng vào Redis để truy xuất lại lịch sử khi cần xử lý AI. Mỗi user có một danh sách riêng biệt.

**Cấu hình:**



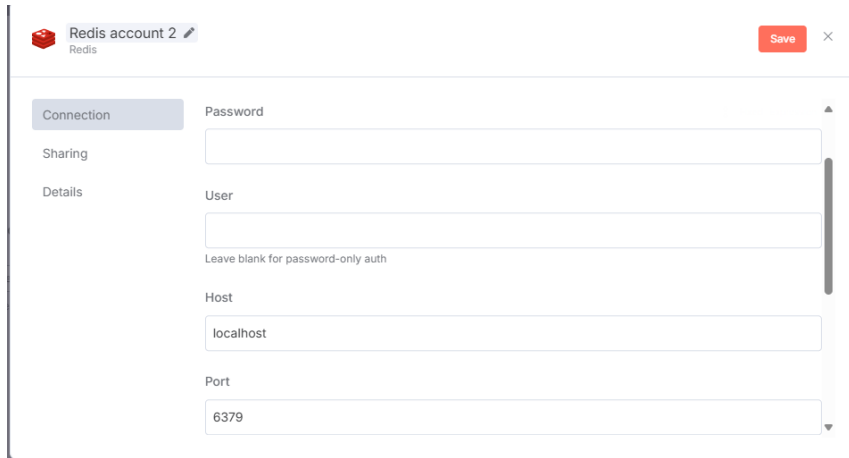


- Credential to connect with: Chọn tài khoản Redis để kết nối. Trong ví dụ là "Redis account".  
Nếu chưa có, chọn "Create new credential".

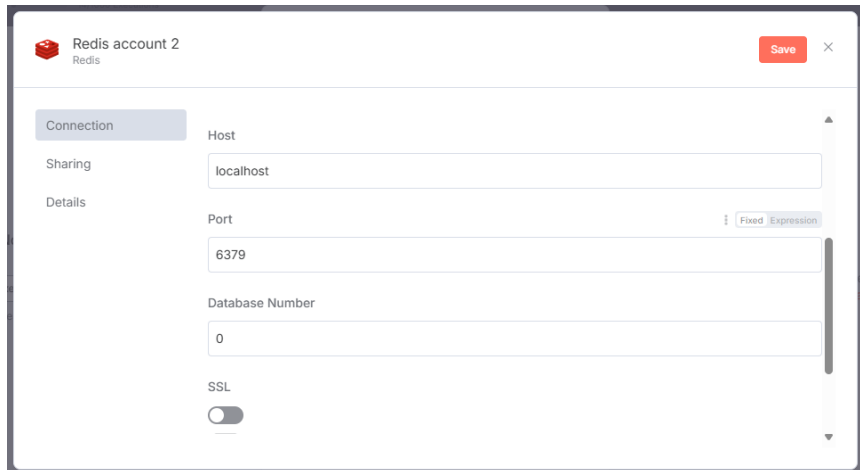


Redis account: Tên credential Redis đã tạo.

Create new credential: Nhấn vào để tạo mới tài khoản kết nối Redis.



The screenshot shows a web interface for configuring a Redis account. The title bar says "Redis account 2" with a Redis logo and a "Save" button. On the left, there are tabs: "Connection" (selected), "Sharing", and "Details". The "Connection" tab contains the following fields: "Password" (empty), "User" (empty), a note "Leave blank for password-only auth", "Host" (filled with "localhost"), and "Port" (filled with "6379").



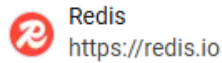
The screenshot shows the same web interface, but now the "Database Number" field is visible and filled with "0". There is also an "SSL" toggle switch which is currently turned off. The "Host" and "Port" fields remain "localhost" and "6379" respectively. The "Connection" tab is still selected.

### **\*\*Thiết lập kết nối Redis\*\***

- Password: Mật khẩu để xác thực với Redis
- User: Tên người dùng Redis (chỉ cần trong Redis ACL – nếu Redis sử dụng xác thực theo user). Để trống nếu Redis chỉ yêu cầu password.
- Host: localhost – Redis chạy cục bộ trên cùng server với n8n.
- Port: 6379 – cổng mặc định Redis.
- Database Number: 0 – chọn DB số 0 trong Redis.
- SSL: tắt – không dùng kết nối bảo mật (chỉ nên dùng cho Redis cục bộ).
- Operation: Push – dữ liệu sẽ được đẩy vào cuối danh sách Redis (FIFO).
- List: {{ \$json.ID }} – tên danh sách là ID người dùng Telegram.
- Data: {{ \$json.messaging }} – nội dung tin nhắn của khách hàng.
- Tail: true – thêm dữ liệu vào cuối danh sách.

### **\* Hướng dẫn tạo tài khoản Redis:**

Bước 1: Truy cập vào trang Redis



### Redis - The Real-time Data Platform

Redis Cloud Fully managed and integrated with Google Cloud, Azure, and AWS. Redis Software Self-managed software with enterprise-grade compliance and reliability. Redis Open Source In ...

Bước 2: Đăng nhập bằng tài khoản gmail hoặc tài khoản github có sẵn (hoặc đăng kí tài khoản mới)

Bước 3: Tạo database ở Redis (lưu ý sau khi đăng nhập thì sẽ được tạo database miễn phí)

## Create your database

Try Redis for Cache Vector Search Database

Essentials Pro

Free database | No credit card needed

Select cloud vendor

Amazon Web Services

Google Cloud

Microsoft Azure

Region

US East (N. Virginia) us-east-1

Included in your Essentials plan

- ✓ 1 Dedicated database
- ✓ 30MB RAM
- ✓ 30 Connections
- ✓ Redis Stack capabilities

[See more plans](#) [Get started](#)

Bước 4: Nhập các trường còn thiếu ở node Redis trong n8n

- Bấm “Connect” để xem các thông số như User, Host, ...

General

Database name

Oanh-free-db

Subscription ⓘ

[Oanh-free-db](#)

Public endpoint

redis-13160.c124.us-central1-1.gce.redns.redis-cloud.com:13160

Connect using Redis CLI, Client, or Insight

Connect

```
public void run()
{
    var muxer = ConnectionMultiplexe
r.Connect(
    new ConfigurationOptions{
        EndPoints= { {"redis-131
60.c124.us-central1-1.gce.redns.redis-cl
oud.com", 13160} },
        User="default",
        Password="*****"
    }
);
```

Với User = “default”, Host = “redis-131... com”, Port = “13160”

Với Password thì copy phần ở đây

Security

Default user

Additional database users can be defined under Data Access Control

☒ On (Username: "default")

[Read more](#)

Default user password

.....

Copy

#### 4.4 Wait:

##### Chức năng:

Tạm dừng luồng xử lý để đảm bảo các thao tác như lưu Redis và tạo khóa xử lý diễn ra tuần tự, tránh xung đột khi nhận nhiều tin nhắn liên tục.

##### Cấu hình:

The screenshot shows the configuration for a 'Wait' step. At the top, there is a 'Wait' title with a pause icon and an 'Execute step' button. Below the title are three tabs: 'Parameters' (selected), 'Settings', and 'Docs'. The 'Parameters' section contains three fields: 'Resume' with a dropdown menu set to 'After Time Interval', 'Wait Amount' with a text input set to '5.00', and 'Wait Unit' with a dropdown menu set to 'Seconds'. To the right of the 'Wait Unit' dropdown are two buttons: 'Fixed' (selected) and 'Expression'.

- Để như mặc định, không cần chỉnh gì thêm.
- Node này đơn giản chỉ dùng để delay luồng xử lý tiếp theo.

#### 4.5 lock:<userID> (Redis Set)

##### **Chức năng:**

Tạo một khoá tạm thời trong Redis theo từng người dùng nhằm ngăn việc xử lý trùng lặp nếu người dùng gửi nhiều tin nhắn liên tiếp quá nhanh. Đây là một kỹ thuật "locking" đơn giản giúp hệ thống hoạt động ổn định.

##### **Cấu hình:**

lock:<userID> Execute step

**Parameters** Settings Docs

Credential to connect with  
Redis account

Operation  
Set

Key  
fx lock:{{ \$json.ID }}

Value  
true

Key Type  
Automatic

Expire  
☒

TTL  
10

- Credential to connect with: Chọn tài khoản Redis đã tạo sẵn, ví dụ: "Redis account".
- Operation: Set – Thiết lập khóa mới.
- Key: lock:{{ \$json.ID }} -> Tạo khoá Redis có tên bắt đầu bằng "lock:" và nối với ID của người dùng Telegram.
- Value: true – Gán giá trị bất kỳ, ở đây là true.
- Key Type: Automatic – Hệ thống tự hiểu kiểu dữ liệu.
- Expire: Bật (ON) – Cho phép khoá này tự động hết hạn sau thời gian nhất định.
- TTL: 10 – Thời gian sống (Time To Live) của khóa là 10 giây. Sau 10 giây, khóa sẽ tự động bị xoá khỏi Redis.

### Ý nghĩa:

Sau khi người dùng gửi tin nhắn, hệ thống sẽ set một khoá lock:userID để tránh việc xử lý tin nhắn đó bị lặp. Trong 10 giây kế tiếp, nếu người dùng spam nhiều tin, hệ thống sẽ kiểm tra khoá này và chỉ xử lý khi khoá đã hết hạn.

### 4.6 If – Kiểm tra khóa Redis

#### Chức năng:

Kiểm tra xem khóa lock:<userID> đã tồn tại trong Redis hay chưa. Nếu khóa chưa tồn tại, cho phép tiếp tục xử lý. Nếu đã tồn tại, dừng lại – nghĩa là đã có một xử lý đang diễn ra rồi.

#### Cấu hình:

The screenshot shows the configuration for an 'If' step in a workflow editor. The 'Parameters' tab is selected, and the 'Conditions' section contains one condition: `fx {{ $json.lock === undefined }}` followed by the operator 'is equal to' and the value 'true'. Below the condition is an 'Add condition' button. The 'Convert types where required' toggle is turned on. The 'Options' section shows 'No properties' and an 'Add option' button.

- Left Value: `{{ $json.lock === undefined }}` -> Biểu thức logic kiểm tra nếu khóa không tồn tại, tức Redis chưa lưu lock:userID.
- Operator: equals
- Right Value: true -> Nếu điều kiện đúng (khóa chưa tồn tại), thực hiện các bước tiếp theo.

- Add condition: Có thể thêm những trường hợp khác với điều kiện là AND hoặc OR

Conditions

fx	{{ \$json.lock === undefined }}	A is equal to	true
AND ^			
		A is equal to	value2

Add condition

Convert types where required

### Luồng xử lý:

- Nhánh true: Tiếp tục lấy lịch sử tin nhắn từ Redis, gộp tin nhắn, gửi sang AI xử lý.
- Nhánh false: Dẫn đến node No Operation, bỏ qua xử lý vì đã có một luồng khác đang làm việc.

### 4.7 GetID (Redis Get)

#### Chức năng:

Lấy lại toàn bộ lịch sử tin nhắn của người dùng từ Redis, dùng để gộp và gửi sang AI xử lý hội thoại theo ngữ cảnh đầy đủ.

#### Cấu hình:



The screenshot shows the 'Parameters' tab of a configuration tool. At the top left is the 'GetID' logo. At the top right is a red button labeled 'Execute step'. Below the logo are two tabs: 'Parameters' (active) and 'Settings'. To the right of the tabs is a 'Docs' link with an external icon. The main configuration area contains several fields: 'Credential to connect with' is a dropdown menu with 'Redis account' selected; 'Operation' is a dropdown menu with 'Get' selected; 'Name' is a text input field containing 'propertyName'; 'Key' is a text input field with a 'fx' icon on the left and a copy icon on the right, containing the expression '{{ \$json.ID }}'; 'Key Type' is a dropdown menu with 'List' selected. Below these fields is an 'Options' section with the text 'No properties' and an 'Add option' button.

- Credential to connect with: Chọn tài khoản Redis đã kết nối (ví dụ: "Redis account").
- Operation: Get – Lấy dữ liệu từ Redis.
- Name: propertyName – tên biến nội bộ mặc định (không cần chỉnh).
- Key: `{{ $json.ID }}` -> Sử dụng ID người dùng Telegram làm key để truy xuất đúng danh sách tin nhắn tương ứng.
- Key Type: List -> Vì Redis đang lưu từng tin nhắn trong danh sách (list) theo user, nên cần chọn kiểu dữ liệu là List.

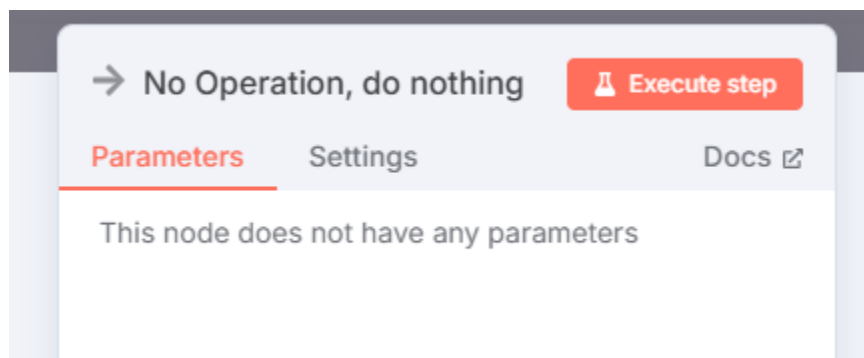
**Ý nghĩa:**

Node này sẽ truy xuất toàn bộ các tin nhắn mà user đã gửi trước đó (được lưu trong node Redis (Push)). Những tin này sẽ được dùng để gộp thành đoạn hội thoại hoàn chỉnh, giúp AI hiểu toàn bộ ngữ cảnh trước khi phản hồi.

#### 4.8 No operation

**Chức năng:**

Node này được dùng để kết thúc nhánh điều kiện không thực hiện gì cả. Trong luồng xử lý chatbot, node này nằm ở nhánh false của node If (khi khóa Redis đã tồn tại), nhằm bỏ qua việc xử lý trùng lặp mà không làm ảnh hưởng đến hệ thống.

**Cấu hình:**

Không có tham số cấu hình. Node này không thực hiện bất kỳ hành động nào. Nó đơn giản là một điểm dừng hoặc “placeholder”.

**Trường hợp sử dụng:**

- Khi bạn muốn tạo một nhánh rẽ để “dừng xử lý”, không chạy tiếp các bước bên dưới.
- Đặc biệt hữu ích trong các tình huống như:
  - Phát hiện đang có xử lý song song → không xử lý tiếp.
  - Điều kiện sai → không làm gì cả nhưng vẫn phải có node để kết thúc hợp lệ.

#### 4.9 deleteID (Redis Delete)

**Chức năng:**

Xóa khóa Redis đã được tạo ở bước trước đó (node lock:<userID>), nhằm mở khóa cho người dùng đó để cho phép xử lý tin nhắn mới trong tương lai.

**Cấu hình:**

The screenshot shows the 'deletelD' interface with three tabs: 'Parameters' (selected), 'Settings', and 'Docs'. A red 'Execute step' button is in the top right. Under 'Parameters', there are three fields: 'Credential to connect with' set to 'Redis account', 'Operation' set to 'Delete', and 'Key' set to a JSONPath expression: `fx {{ $('lock:<userID>').item.json.ID }}`.

- Credential to connect with: "Redis account"
- Operation: Delete – thao tác xóa khóa Redis.
- Key: `={{ $('lock:<userID>').item.json.ID }}` -> Lấy ID từ node đặt khóa trước đó (lock:<userID>), và ghép thành khóa Redis cần xóa: lock:<ID>.

### Ý nghĩa:

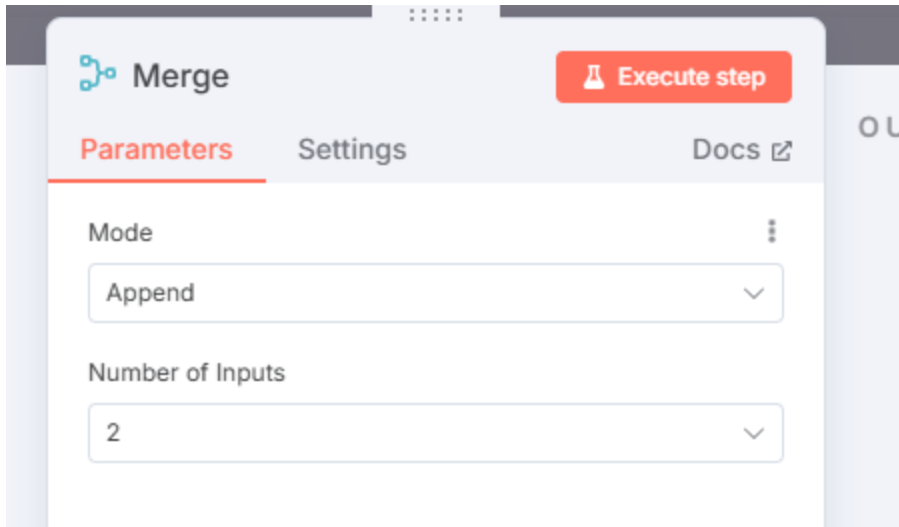
Trước đó, khi người dùng gửi tin nhắn, hệ thống đã đặt một khóa tạm (lock:ID) để ngăn xử lý song song. Sau khi lấy xong toàn bộ tin nhắn từ Redis (GetID) và chuẩn bị gửi sang AI, hệ thống phải xóa khóa này để cho phép xử lý tiếp theo nếu có tin mới từ người dùng. Nếu không xóa, các lần nhấn tiếp theo sẽ luôn bị node If chặn lại vì khóa vẫn còn tồn tại.

### 4.10 Merge

#### Chức năng:

Gộp dữ liệu đầu vào từ 2 nhánh riêng biệt là node Edit Fields1 (chứa ID người dùng, username...) và Redis node GetID (lấy lịch sử tin nhắn Redis) thành 1 luồng dữ liệu thống nhất, để sử dụng ở bước xử lý tiếp theo như gộp nội dung, gửi AI...

#### Cấu hình:

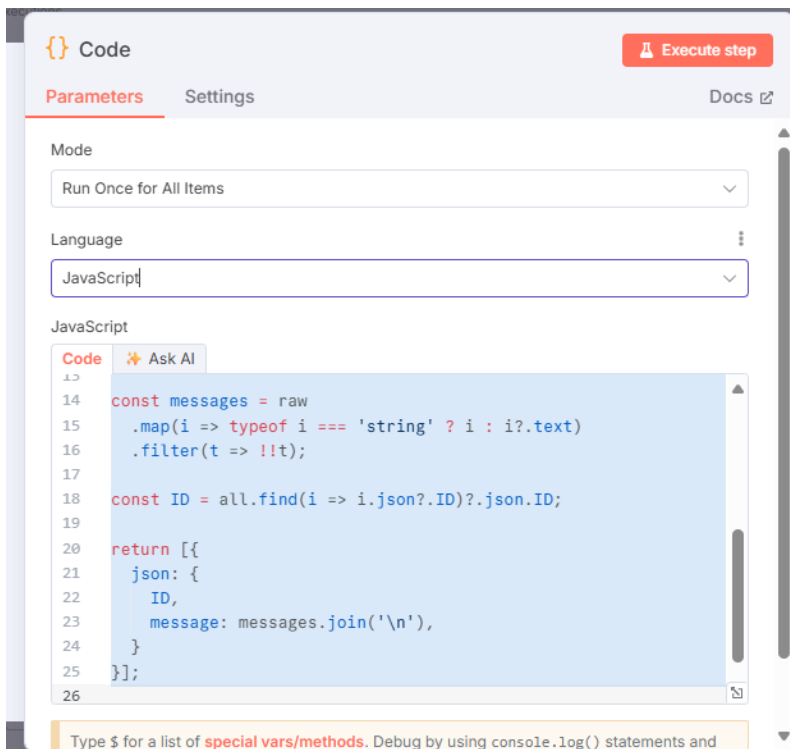


- Mode: Append -> Gộp tất cả dữ liệu đầu vào thành một mảng gồm nhiều object.
- Number of Inputs: 2 -> Vì đang gộp 2 nhánh dữ liệu.

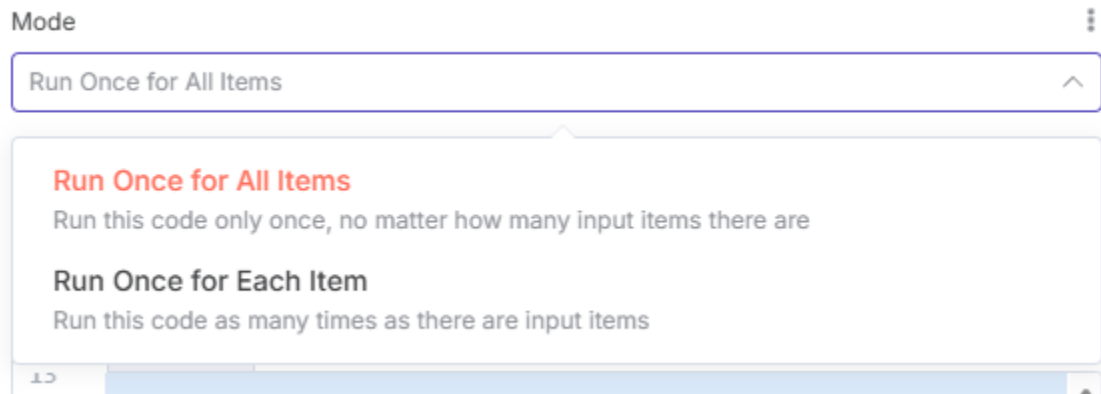
#### 4.11 Code (Ghép chuỗi hội thoại)

**Chức năng:** Ghép toàn bộ tin nhắn trước đó thành một chuỗi văn bản thống nhất để gửi cho AI xử lý tư vấn theo ngữ cảnh.

**Cấu hình:**



- Mode:

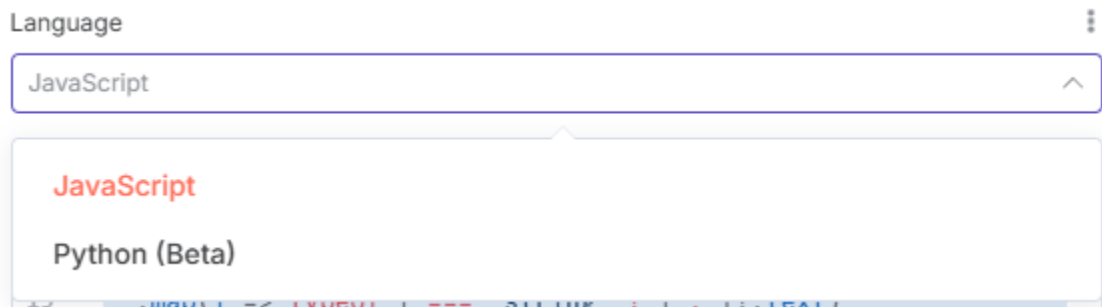


Run Once for All Items: Chạy đoạn code một lần duy nhất, dù có 1 hay nhiều item đầu vào.

Run Once for Each Item: Chạy đoạn code lặp lại cho từng dòng dữ liệu đầu vào.

Trong workflow chatbot, tôi chọn Run Once for All Items vì đang lưu nhiều tin nhắn của cùng một người dùng (ID Telegram) vào Redis → rồi dùng node Code để ghép các tin nhắn đó lại thành một đoạn hội thoại gửi cho AI xử lý.

- Language: JavaScript  
Có 2 loại ngôn ngữ là JavaScript và Python



- Mục tiêu: xử lý dữ liệu đầu vào và tạo một chuỗi tin nhắn để truyền cho node AI

### Code:

`const input = items[0];` → lấy dòng dữ liệu đầu tiên (thường là tin nhắn lịch sử từ Redis).

`const all = $input.all();` → lấy toàn bộ các dòng đầu vào từ node Merge (bao gồm cả thông tin ID từ nhánh còn lại).

```
const fieldName = Object.keys(input.json).find(k => {  
  const val = input.json[k];  
  return typeof val === 'string' || Array.isArray(val);  
});
```

 → tìm trường dữ liệu chứa chuỗi hoặc mảng chuỗi để xử lý

`let raw = input.json[fieldName];` → chuẩn hóa dữ liệu thành mảng

`if (typeof raw === 'string') raw = [raw];`

`if (!Array.isArray(raw)) raw = [];`

(raw: chứa danh sách các đoạn tin nhắn. Nếu là string đơn → đưa vào mảng. Nếu không phải string hay mảng → đưa về mảng rỗng để tránh lỗi)

`const messages = raw.map(i => typeof i === 'string' ? i : i?.text).filter(t => !!t);` → lọc và giữ lại những đoạn text hợp lệ

`const ID = all.find(i => i.json?.ID)?.json.ID;` → lấy lại ID người dùng để truyền theo

`return [{`

`json: {`

`ID,`

`message: messages.join('\n'),`

`}`

`}];` → trả về một object gồm ID và chuỗi message gộp

#### 4.12 Kiểm tra "chốt đơn" (If)

**Chức năng:** Phát hiện khi người dùng gửi tin nhắn "chốt đơn" để kích hoạt quá trình xử lý đơn hàng.

### Cấu hình:

**Kiểm tra chốt đơn** Execute step

**Parameters** Settings Docs

**Conditions**

fx	{{ (\$json.message    '').toLowerCase() }}	A contains
fx	chốt đơn	

Add condition

Convert types where required ☐

**Options**

No properties

Add option

- Left Value: `{{ ($json.message || '').toLowerCase() }}`
- Right Value: chốt đơn
- Operator: contains

### Luồng xử lý:

- Nhánh true: Nếu khách hàng có nhấn "chốt đơn" → hệ thống tiếp tục xử lý đơn hàng: Truy xuất thông tin đơn hàng; Tách tên, địa chỉ, sản phẩm; Lưu đơn hàng và gửi xác nhận.
- Nhánh false: Nếu không nhắc tới "chốt đơn": Bot sẽ xử lý như cuộc trò chuyện bình thường. Có thể gửi phản hồi từ AI Agent hoặc tiếp tục ghi nhận thêm nội dung khác.

#### 4.13 Lấy đơn hàng (Redis Get)

**Chức năng:** Truy xuất tin nhắn xác nhận đơn hàng đã lưu trong Redis từ phản hồi AI để phân tích.

**Cấu hình:**

**Lấy đơn hàng** Execute step

**Parameters** Settings Docs

Credential to connect with  
Redis account

Operation  
Get

Name  
propertyName

Key  
fx {{ 'last\_confirm\_message:' + \$json.ID }}

Key Type  
Automatic

Options  
No properties  
Add option

- Credential to connect with: Redis account – chọn tài khoản Redis đã thiết lập.
- Operation: Get – thao tác đọc dữ liệu từ Redis.
- Name: propertyName – tên thuộc tính lưu output (không bắt buộc).



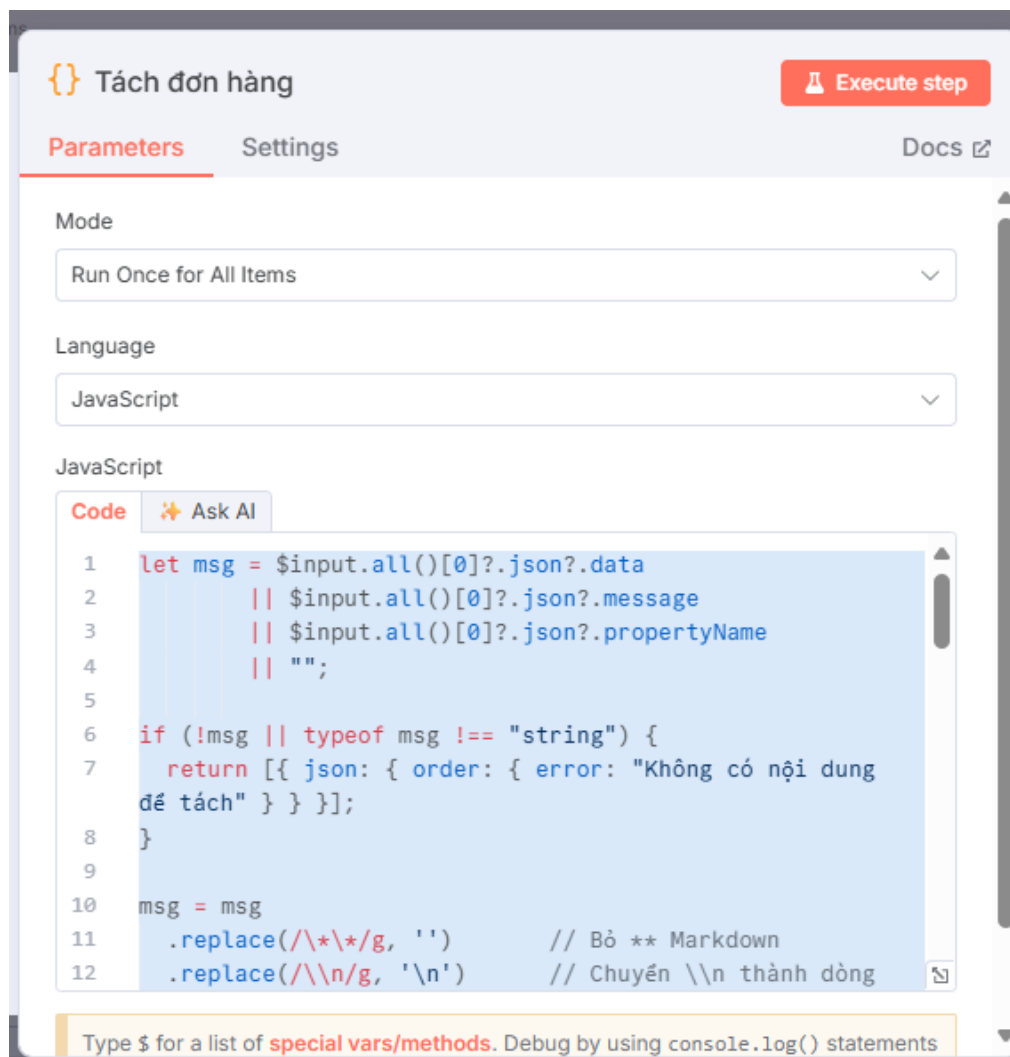
- Key: `={{ 'last_confirm_message:' + $json.ID }}` → Tạo khóa động dựa theo ID Telegram người dùng.
- Key Type: Automatic – để mặc định, Redis tự chọn kiểu dữ liệu.

**Ý nghĩa:** Redis được dùng như một bộ nhớ tạm (cache), giúp lưu lại phản hồi của AI cho từng người dùng. Khi người dùng xác nhận "chốt đơn", hệ thống có thể lấy lại chính xác nội dung đơn hàng mà AI đã đề xuất trước đó, đảm bảo không bị mất ngữ cảnh hay thông tin.

#### 4.14 Tách đơn hàng (Code)

**Chức năng:** Tách các trường như tên khách hàng, sản phẩm, size, màu sắc, số lượng, địa chỉ và số điện thoại từ phản hồi xác nhận đơn hàng của AI.

**Cấu hình:**



- Mode: Run Once for All Items
- Language: JavaScript

**Code:**

```
let msg = $input.all()[0]?.json?.data
```

```
|| $input.all()[0]?.json?.message
```

```
|| $input.all()[0]?.json?.propertyName
```

|| ""; -> Dò tìm nội dung đơn hàng từ 3 nguồn phổ biến: data, message, hoặc propertyName. Nếu không có, gán chuỗi rỗng "").

```
if (!msg || typeof msg !== "string") {
```

```
  return [{ json: { order: { error: "Không có nội dung để tách" } } }];
```

} -> Nếu không có nội dung hoặc không phải chuỗi → trả lỗi JSON "Không có nội dung để tách".

```
msg = msg
```

```
.replace(/\*/g, "")
```

```
.replace(/\n/g, '\n')
```

```
.replace(/\r/g, "")
```

```
.trim(); -> loại bỏ các kí tự thừa
```

```
let order = {
```

```
  Ten: "",
```

```
  SDT: "",
```

```
  DiaChi: "",
```

```
  SanPham: "",
```

```
  SoLuong: "",
```

Size: "",

MauSac: "",

isChot: true

}; -> Khởi tạo các biến lưu trữ order lưu thông tin sau khi tách. isChot: true ngằm hiểu đơn đã được xác nhận và sẵn sàng xử lý.

const lines = msg.split('\n').map(line => line.trim()).filter(line => line); -> Cắt đoạn văn thành từng dòng. Loại bỏ dòng trống và khoảng trắng thừa.

```
for (const line of lines) {
```

```
  if (line.startsWith("- Tên:")) {
```

```
    order.Ten = line.substring(6).trim();
```

```
  } else if (line.startsWith("- Sản phẩm:")) {
```

```
    const fullSanPham = line.substring(11).trim();
```

```
    const match = fullSanPham.match(/^(.+?)\s*(\/);
```

```
    order.SanPham = match ? match[1].trim() : fullSanPham;
```

```
    const sizeMatch = fullSanPham.match(/size\s*([A-Z0-9]+)/i);
```

```
    if (sizeMatch) order.Size = sizeMatch[1].trim();
```

```
    const colorMatch = fullSanPham.match(/màu\s+([^\,]+)/i);
```

```
    if (colorMatch) order.MauSac = colorMatch[1].trim();
```

```
    const quantityMatch = fullSanPham.match(/(?:\d+)\s*chiếc/i);
```

```
    if (quantityMatch) order.SoLuong = quantityMatch[1].trim();
```

```
  } else if (line.startsWith("- Địa chỉ:")) {
```

```
    order.DiaChi = line.substring(10).trim();
```

```
  } else if (line.startsWith("- Số điện thoại:")) {
```

```

    order.SDT = line.substring(16).trim();
}
}

```

return [{ json: { order } }]; → Trả về đối tượng JSON chứa toàn bộ thông tin đơn hàng đã tách được.

#### 4.15 Truy xuất giá (Google Sheets)

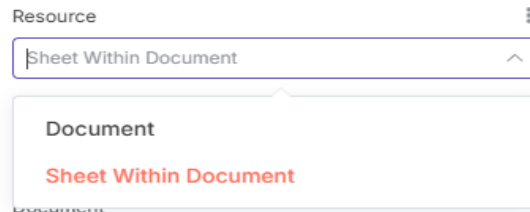
**Chức năng:** Lấy giá sản phẩm theo tên và size từ Google Sheet chứa bảng ProductInfo.

**Cấu hình:**

The screenshot shows the 'Truy xuất giá' (Price Lookup) configuration window. It includes a red 'Execute step' button at the top right. The 'Parameters' tab is active, showing the following settings:

- Credential to connect with:** Google Sheets account
- Resource:** Sheet Within Document
- Operation:** Get Row(s)
- Document:** From list, Thông tin CSKH
- Sheet:** From list, ProductInfo
- Filters:**
  - Column:** Loading options...
  - Value:** `fx {{ $json.order.SanPham }}`

- Credential to connect with: Google Sheets account – chọn tài khoản đã kết nối với Google API.
- Resource:



Sheet Within Document – thao tác trực tiếp trên sheet trong tài liệu Google Sheet.

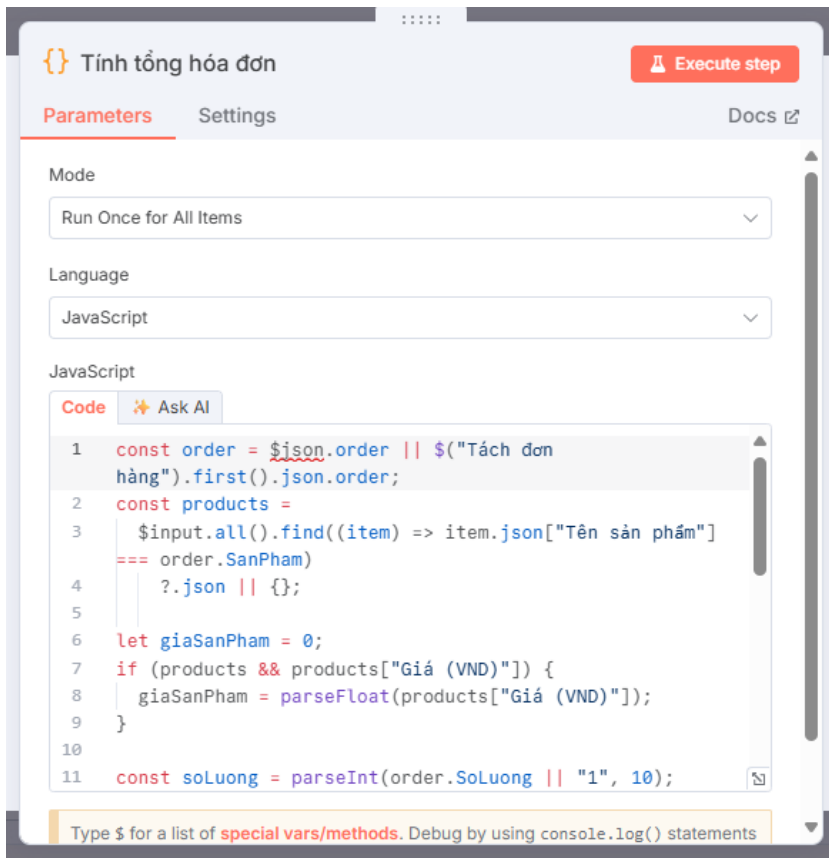
Document – lấy toàn bộ sheet

- Operation: Get Row(s) – lấy dòng dữ liệu phù hợp với điều kiện lọc.
- Document: Thông tin CSKH – tên file Google Sheet lưu trữ dữ liệu sản phẩm.
- Sheet: ProductInfo – tên sheet bên trong chứa bảng giá sản phẩm.
- Filter: (Lấy 2 dữ liệu để truy xuất giá là “Tên sản phẩm” và “Size”)
  - Column: Bạn sẽ chọn tên cột để so sánh -> chọn cột "Tên sản phẩm"
  - Value = {{ \$json.order.SanPham }} -> → Lấy tên sản phẩm từ đơn hàng đã được tách ở node trước (order.SanPham). Đây là giá trị để so sánh với nội dung trong cột đã chọn bên trên.

#### 4.16 Tính tổng hóa đơn:

**Chức năng:** Tính toán **tổng số tiền đơn hàng**, dựa vào: Giá sản phẩm, số lượng đặt, phí ship (nếu có). Sau đó trả về toàn bộ thông tin đơn hàng đã chuẩn hóa để lưu hoặc xác nhận.

**Cấu hình:**



- Mode: Run Once for All Items
- Language: JavaScripts

### Code:

`const order = $json.order || $("Tách đơn hàng").first().json.order;` -> Nếu order đã có trong node hiện tại → dùng luôn. Nếu không có → lấy từ node Tách đơn hàng.

`const products =`

`$input.all().find((item) => item.json["Tên sản phẩm"] === order.SanPham)`

`?.json || {};` -> Tìm dòng dữ liệu sản phẩm từ Google Sheets có "Tên sản phẩm" trùng với tên sản phẩm khách đặt. Nếu không tìm thấy thì products là {}.

`let giaSanPham = 0;`

`if (products && products["Giá (VND)"]) {`

```
giaSanPham = parseFloat(products["Giá (VND)"]);
```

} -> Lấy giá từ cột "Giá (VND)", chuyển thành số thập phân (parseFloat) để tính toán.

```
const soLuong = parseInt(order.SoLuong || "1", 10); -> Lấy số lượng sản phẩm đặt hàng. Mặc định là 1 nếu không có giá trị.
```

```
const phiShip = 30000;
```

```
const tongTien = giaSanPham * soLuong + phiShip;
```

```
return [
```

```
{
```

```
  json: {
```

```
    Ten: order.Ten,
```

```
    SDT: order.SDT,
```

```
    DiaChi: order.DiaChi,
```

```
    SanPham: order.SanPham,
```

```
    Size: order.Size,
```

```
    MauSac: order.MauSac,
```

```
    SoLuong: soLuong,
```

```
    TongTien: tongTien,
```

```
  },},]; -> Trả về thông tin đầy đủ đơn hàng, bao gồm cả TongTien, để gửi xác nhận và lưu vào Google Sheets.
```

#### *4.17 Kiểm tra chốt đơn1 (If)*

**Chức năng:** Kiểm tra nếu đơn hàng có isChot == true (đã đủ thông tin), sẽ tiếp tục xử lý đơn hàng.

**Cấu hình:**

**Kiểm tra chốt đơn** Execute step

**Parameters** Settings Docs

Conditions

fx	{{ \$("Tách đơn hàng").item.json.order.isChot }}	<input checked="" type="checkbox"/> is equal to	▼
	ot }}		
	true		▼

Add condition

Options

No properties

Add option ▼

- Left Value: `{{ $("Tách đơn hàng").item.json.order.isChot }}`
- Operator: equals
- Right Value: true

**Ý nghĩa:** Khi AI phản hồi nội dung đơn hàng (tên, số, sản phẩm...), node Tách đơn hàng sẽ đánh dấu `isChot = true` nếu đủ thông tin. Node này đóng vai trò như cổng kiểm tra cuối cùng để đảm bảo hệ thống chỉ lưu đơn hàng đã xác nhận rõ ràng.

#### Luồng xử lý:

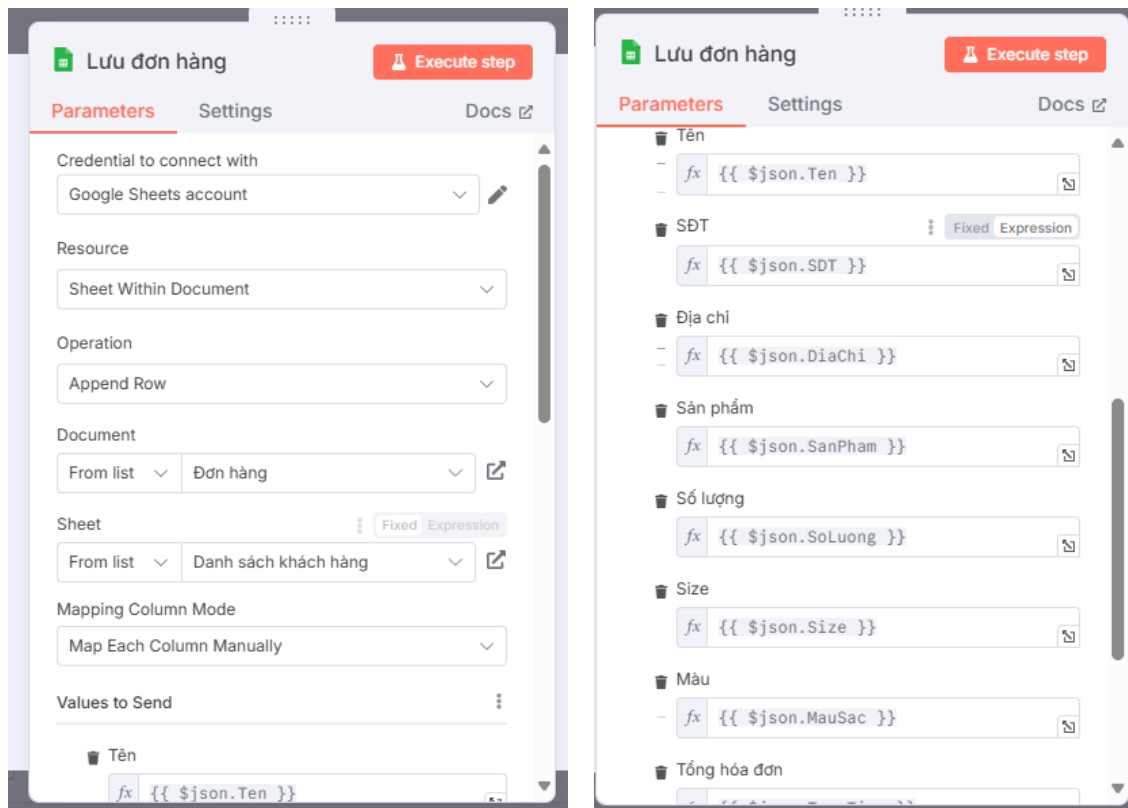
- Nhánh true: Chuyển sang Lưu đơn hàng và Gửi xác nhận.
- Nhánh false: Không thực hiện lưu đơn, có thể kết thúc hoặc thông báo lỗi thiếu thông tin.

#### 4.18 Lưu đơn hàng

**Chức năng:** Ghi thông tin đơn hàng vào Google Sheets, để lưu trữ tất cả các đơn khách đã đặt. Mỗi lần ghi là một dòng mới tương ứng một đơn hàng.



## Cấu hình:



- Credential to connect with: Tài khoản Google Sheets đã kết nối (VD: Google Sheets account).
- Resource: Sheet Within Document – Làm việc trực tiếp trên bảng tính cụ thể.
- Operation: Append Row – Ghi thêm một dòng mới vào cuối bảng.
- Document: Đơn hàng – Tên file Google Sheets chứa dữ liệu đơn hàng.
- Sheet: Danh sách khách hàng – Tên sheet bên trong file Đơn hàng.
- Mapping Column Mode: Map Each Column Manually – Cho phép bạn gán từng cột thủ công tương ứng với biến dữ liệu.
- Values to Send – Gán dữ liệu từng cột:
  - Tên : {{ \$json.Ten }}
  - SĐT: {{ \$json.SDT }}
  - Địa chỉ: {{ \$json.DiaChi }}
  - Sản phẩm: {{ \$json.SanPham }}
  - Số lượng: {{ \$json.SoLuong }}

- Size: {{ \$json.Size }}
- Màu: {{ \$json.MauSac }}
- Tổng hóa đơn: {{ \$json.TongTien }}

**Ý nghĩa:** Tự động lưu lại toàn bộ đơn hàng sau khi đã xác nhận và tính toán.

#### 4.19 Gửi xác nhận (Telegram Trigger)

**Chức năng:** Sau khi lưu thành công đơn hàng, hệ thống sẽ gửi tin nhắn xác nhận lại cho khách hàng qua Telegram, bao gồm đầy đủ thông tin đơn hàng vừa đặt.

**Cấu hình:**

**Gửi xác nhận** Execute step

**Parameters** Settings Docs

Credential to connect with  
Telegram account

Resource  
Message

Operation  
Send Message

Chat ID  
fx {{ \$('Wait').item.json.ID }}

Text  
Cảm ơn {{ \$json.Ten }}, đơn hàng của bạn đã được ghi nhận!  
fx Sản phẩm: {{ \$json.SanPham }}  
Size: {{ \$json.Size }}  
Số lượng: {{ \$json.SoLuong }}  
Địa chỉ: {{ \$json.DiaChi }}

Reply Markup  
None

- Credential to connect with: Tài khoản Telegram đã kết nối (Telegram account).
- Resource: Message – Loại tài nguyên là tin nhắn.
- Operation: Send Message – Gửi tin nhắn đến người dùng.

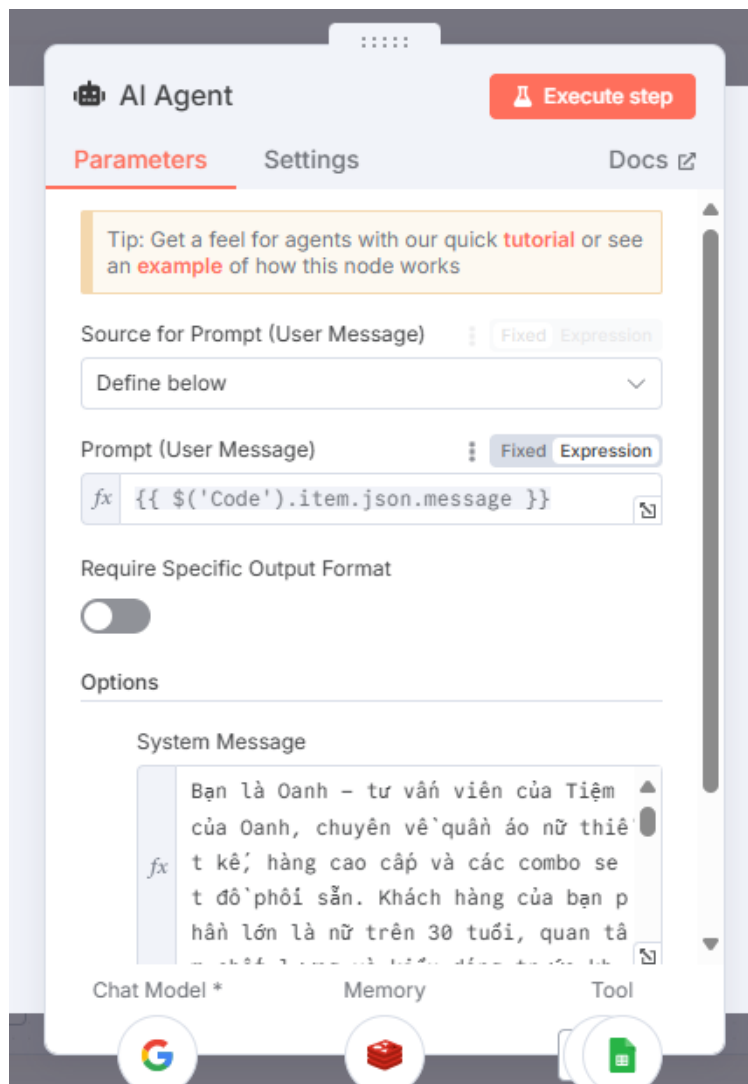
- Chat ID: `{{ $('Wait').item.json.ID }}` -> Lấy lại ID của người dùng từ node Wait – dùng để gửi đúng về cho người đã đặt hàng.
- Text: Nội dung phản hồi gửi lại khách (sử dụng biến động để cá nhân hóa)
- Reply Markup: None – Không sử dụng giao diện trả lời đặc biệt.

**Ý nghĩa:** Giúp khách xác nhận lại rằng thông tin đã đúng, có thể phản hồi nếu cần sửa.

#### 4.20 AI Agent:

**Chức năng:** Là node chính dùng để xử lý ngữ nghĩa của tin nhắn người dùng. Hoạt động như một chatbot tư vấn, có thể sử dụng context từ cơ sở dữ liệu (Memory), thông tin sản phẩm (Products), FAQ để phản hồi. Tích hợp trí tuệ nhân tạo (AI) để hiểu và phản hồi dựa trên yêu cầu.

**Cấu hình:**



- Source for Prompt (User Message): Define below -> Tin nhắn người dùng được truyền trực tiếp từ node trước thông qua biểu thức tùy chỉnh.
- Prompt (User Message): {{ \$('Code').item.json.message }} -> Lấy nội dung tin nhắn từ node có tên Code, trong trường hợp này là item.json.message.
- System Message: Bạn là Oanh – tư vấn viên của Tiệm của Oanh, chuyên về quần áo nữ thiết kế... -> Thiết lập vai trò cho AI là một tư vấn viên thời trang, giúp cá nhân hóa trải nghiệm tư vấn cho khách hàng.
- Chat Model: Google Gemini Chat Model -> Sử dụng mô hình Gemini (của Google) để xử lý ngôn ngữ tự nhiên.
- Memory: Redis Chat Memory: lưu lại trạng thái cuộc hội thoại.
- Tools: Products và FAQs: là Google Sheets chứa thông tin sản phẩm và câu hỏi thường gặp.

**Ý nghĩa:** Node AI Agent đóng vai trò trung tâm của workflow chatbot, tiếp nhận tin nhắn người dùng, tham chiếu các dữ liệu hỗ trợ (sản phẩm, FAQ, trạng thái hội thoại) để tạo phản hồi phù hợp và tự nhiên. Từ đó giúp AI có thể "tư vấn như người thật", đồng thời ghi nhớ thông tin đơn hàng và gửi trả lại kết quả cho khách qua Telegram.

#### 4.21 Google Sheets Tool – Products

**Chức năng:** Truy vấn thông tin sản phẩm từ bảng Google Sheets chứa dữ liệu sản phẩm. Đọc toàn bộ danh sách sản phẩm từ Google Sheets để chatbot có thể tư vấn thông tin chi tiết về sản phẩm cho khách hàng.

##### Cấu hình:

- Credential to connect with: Tài khoản Google Sheets đã kết nối (VD: Google Sheets account).
- Tool Description: Set Automatically – Không cần chỉnh thủ công mô tả công cụ.
- Resource: Sheet Within Document – Làm việc với một sheet cụ thể trong file Google Sheets.
- Operation: Get Row(s) – Lấy dữ liệu từ nhiều dòng trong sheet
- Document: Thông tin sản phẩm – Tên file Google Sheets chứa danh sách sản phẩm.
- Sheet: Sản phẩm – Tên sheet chứa dữ liệu về các sản phẩm.
- Filters: Không thiết lập – Chưa lọc dữ liệu, tức là lấy toàn bộ dòng sản phẩm.

**Products** Execute step

**Parameters** Settings Docs

Credential to connect with **AI Agent**  
Google Sheets account

Tool Description  
Set Automatically

Resource  
Sheet Within Document

Operation  
Get Row(s)

Document  
From list Thông tin CSKH

Sheet  
From list ProductInfo

Filters  
Currently no items exist  
Add Filter

**Ý nghĩa:** Giúp chatbot nắm được thông tin sản phẩm để hỗ trợ tư vấn như tên, giá, mô tả, size, màu,...

#### 4.22 Google Sheets Tool – FAQs

**Chức năng:** Truy vấn câu hỏi thường gặp (FAQs) để AI phản hồi những câu như: "giặt sao?", "chất vải gì?", "phối đồ sao?" Đọc dữ liệu câu hỏi thường gặp từ Google Sheets để chatbot có thể tự động phản hồi các thắc mắc phổ biến của khách hàng

**Cấu hình:**

- Credential to connect with: Tài khoản Google Sheets đã kết nối (Google Sheets account).
- Tool Description: Set Automatically – Không cần chỉnh thủ công mô tả công cụ.
- Resource: Sheet Within Document – Làm việc với một sheet cụ thể trong file Google Sheets.
- Operation: Get Row(s) – Lấy dữ liệu từ nhiều dòng trong sheet

- Document: Câu hỏi thường gặp – Tên file chứa dữ liệu FAQ.
- Sheet: FAQs – Tên sheet chứa danh sách các câu hỏi và câu trả lời.
- Filters: Không thiết lập – Chưa lọc dữ liệu, tức là lấy toàn bộ dòng sản phẩm.

**Ý nghĩa:** Giúp chatbot tìm kiếm và đưa ra câu trả lời phù hợp cho các câu hỏi như: *“Chính sách đổi trả thế nào?”*, *“Shop có miễn phí vận chuyển không?”*,...

#### 4.23 Lưu trữ đơn hàng (Redis Set)

**Chức năng:** Lưu nội dung phản hồi của AI (dạng xác nhận đơn hàng) vào Redis để sử dụng lại nếu cần xử lý đơn sau này. Cho phép truy xuất đơn hàng gần nhất dựa trên ID người dùng Telegram.

## Cấu hình:

The screenshot shows a configuration window for a Redis step. At the top, there's a title bar with a red cube icon, the text 'Lưu trữ đơn hàng' (Save order), and a red 'Execute step' button. Below the title bar are three tabs: 'Parameters' (selected), 'Settings', and 'Docs'. The 'Parameters' tab contains several fields: 'Credential to connect with' (a dropdown menu showing 'Redis account'), 'Operation' (a dropdown menu showing 'Set'), 'Key' (a text input field with a formula: `{{ 'last_confirm_message:' + $json.ID }}`), 'Value' (a text input field with a formula: `{{ $json.output }}`), 'Key Type' (a dropdown menu showing 'Automatic'), and 'Expire' (a toggle switch that is currently turned off).

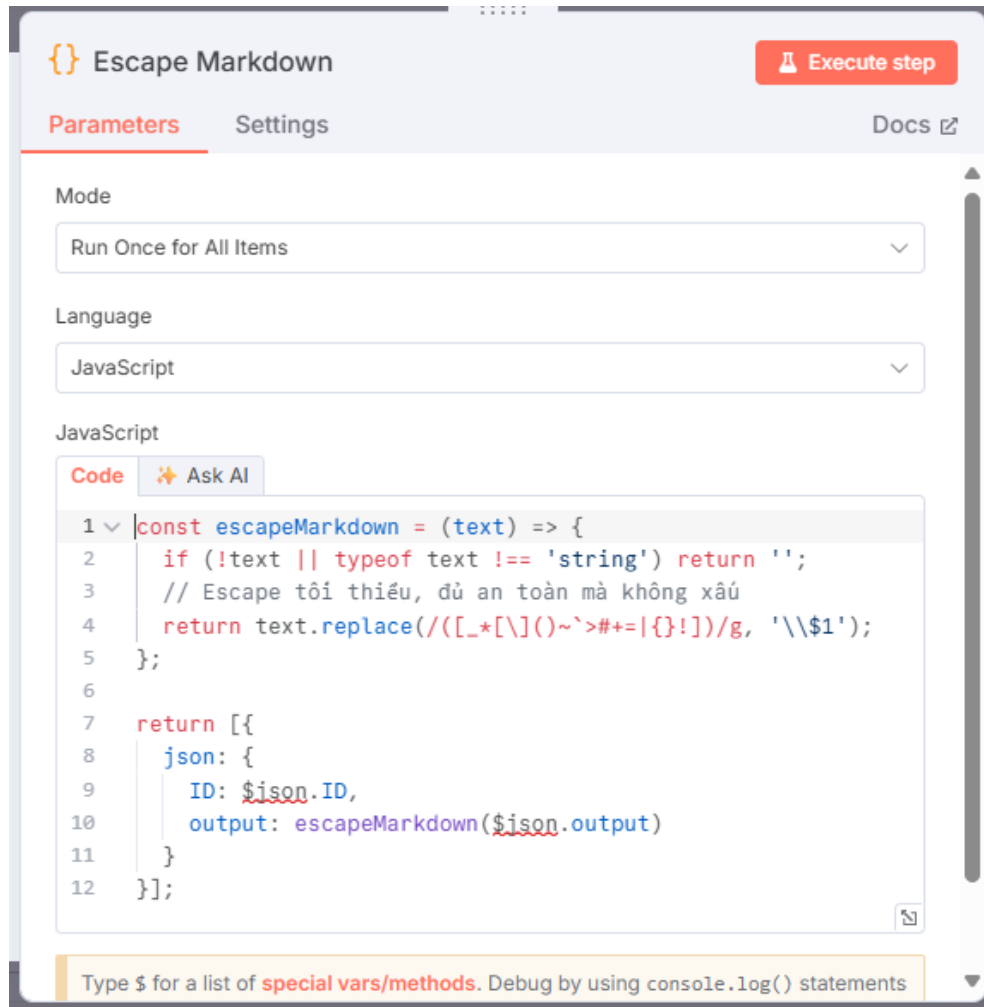
- Credential to connect with: Redis account -> Kết nối với tài khoản Redis đã cấu hình từ trước trong n8n (dùng để lưu trữ tạm).
- Operation: Set -> Thiết lập (ghi) một key-value mới vào Redis.
- Key: `{{ 'last_confirm_message:' + $json.ID }}` -> Key được tạo theo format: `last_confirm_message:<Telegram ID>`. Dùng để định danh đơn hàng cuối cùng cho từng người dùng cụ thể. `$json.ID` là ID người dùng Telegram – được truyền từ node trước đó.
- Value: `{{ $json.output }}` -> Đây là nội dung chi tiết của đơn hàng mà AI đã xử lý và chuẩn bị gửi lại. `$json.output` thường chứa đoạn text cuối cùng của AI Agent, ví dụ: đơn hàng đã tổng hợp, thông tin xác nhận, v.v.
- Key Type: Automatic -> Redis tự động xác định kiểu của giá trị lưu (thường là string, JSON...).
- Expire: Tắt (Off) -> Key không bị hết hạn – nghĩa là đơn hàng sẽ được lưu vĩnh viễn trong Redis trừ khi xóa thủ công hoặc ghi đè.

**Ý nghĩa:** Giữ lịch sử đơn hàng tạm thời để dùng cho các bước xử lý tiếp theo (gửi tin nhắn, xác nhận lại...).

#### 4.24 Escape Markdown (Code)

**Chức năng:** Làm sạch ký tự Markdown trong tin nhắn (như \*, \_ [...]) để tránh lỗi định dạng khi gửi Telegram.

**Cấu hình:**



- Mode: Run Once for All Items
- Language: JavaScripts

**Code:**

```
const escapeMarkdown = (text) => {
```

if (!text || typeof text !== 'string') return ''; -> Nếu text là null, undefined, hoặc không phải chuỗi, thì trả về chuỗi rỗng (").



`return text.replace(/([_*\[\]\~`>#+=|{}!])/g, '\\$1');` -> Dùng `.replace()` để thêm dấu gạch chéo (`\`) trước các ký tự đặc biệt có thể gây lỗi định dạng Markdown. Regex `([_*\[\]\~`>#+=|{}!])` sẽ tìm các ký tự Markdown đặc biệt.

`};`

`return [{`

`json: {`

`ID: $json.ID,`

`output: escapeMarkdown($json.output)`

`}`

`}]`; -> trả về một object chứa dữ liệu ở định dạng json gồm: ID: Lấy từ dữ liệu đầu vào (`$json.ID`) – thường là ID người dùng Telegram, output: Chuỗi đã được **escape Markdown**, từ `$json.output`.

**Ý nghĩa:** Đảm bảo an toàn, không bị "vỡ định dạng" khi tin nhắn chứa dấu ngoặc, dấu \*, [ hoặc !]

#### *4.25 Gửi tin nhắn bình thường "Telegram Trigger"*

**Chức năng:** Node này dùng để gửi tin nhắn văn bản đến người dùng Telegram thông qua Telegram Bot API. Nó được sử dụng sau khi nội dung đã được tạo từ AI Agent và đã được xử lý Markdown.

**Cấu hình:**

**Gửi tin nhắn bình thường** Execute step

**Parameters** Settings Docs

Credential to connect with  
Telegram account

Resource  
Message

Operation  
Send Message

Chat ID  
fx {{ \$('Wait').item.json.ID }}

Text  
fx {{ \$json.output }}

Reply Markup  
None

Additional Fields  
No properties

Add Field

- Credential to connect with: Tài khoản Telegram đã kết nối (Telegram account).
- Resource: Message – Loại tài nguyên là tin nhắn.
- Operation: Send Message – Gửi tin nhắn đến người dùng.
- Chat ID: {{ \$('Wait').item.json.ID }} -> Lấy lại ID của người dùng từ node Wait – dùng để gửi đúng về cho người đã đặt hàng.
- Text: {{ \$json.output }} -> Dữ liệu output là kết quả đã được escape Markdown từ node Escape Markdown, đảm bảo không bị lỗi định dạng khi hiển thị trong Telegram.
- Reply Markup: None – Không sử dụng giao diện trả lời đặc biệt.

**Ý nghĩa:** Đảm bảo tin nhắn đến đúng người và đúng định dạng (an toàn với Markdown Telegram). Vẫn có thể tư vấn đến cho khách hàng.

## V. Lưu ý khi sử dụng

- Hạn chế spam tin nhắn, hệ thống có giới hạn khoá tạm thời 10 giây.
- Nếu muốn đặt hàng, khách cần nhấn đúng từ khóa "chốt đơn".
- Nếu chatbot không phản hồi ngay, có thể do đang xử lý một cuộc trò chuyện trước.
- Đảm bảo dữ liệu sản phẩm và FAQs được cập nhật đầy đủ trong Google Sheets.

## VI. Kết luận

Việc áp dụng chatbot trên n8n giúp shop bán áo quần tự động hoá chăm sóc khách hàng, giảm tải cho nhân viên, tăng tốc độ phản hồi và nâng cao trải nghiệm người dùng. Tài liệu này cung cấp toàn bộ thông tin cần thiết để triển khai và vận hành chatbot từ A-Z.

Hệ thống có khả năng mở rộng dễ dàng, có thể tích hợp thêm các kênh mới (Zalo OA, Facebook Messenger...) hoặc nâng cấp mô hình AI trong tương lai.

Nếu có bất kỳ vấn đề hoặc yêu cầu hỗ trợ, vui lòng liên hệ bộ phận kỹ thuật hoặc người phụ trách triển khai chatbot của shop.