

Thư viện tìm hiểu: [node-red-contrib-telegrambot](https://github.com/node-red-contrib-telegrambot)

Mục Lục

I. Giới thiệu:	1
II. Chi tiết các node trong thư viện	1
1. Telegram receiver	1
2. Telegram sender Node	3
3. Command Node	4
4. Event Node	5
5. Telegram Reply	7
III. Kịch bản sử dụng thư viện và cài đặt:	8
1. Kịch bản:	8
2. Cài đặt:	10

I. Giới thiệu:

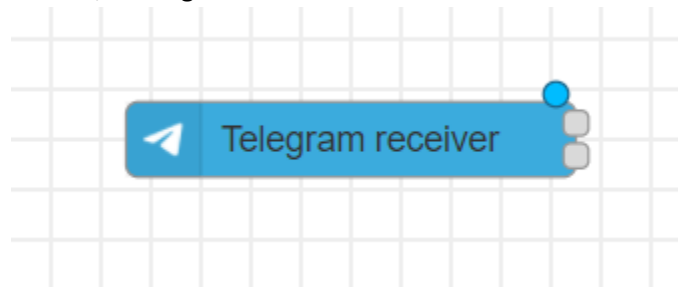
Trong quãng thời gian qua, các mạng xã hội phổ biến như Facebook, Zalo đang dần đánh mất niềm tin của một bộ phận người dùng trong vấn đề bảo mật. Xu hướng tìm kiếm một ứng dụng chat khác tốt hơn là một nhu cầu cần thiết, trong các ứng dụng sáng giá thì có Telegram.

Telegram cho phép chúng ta tạo bot để thực hiện tự động các công việc từ đơn giản đến phức tạp tùy vào nhà phát triển đã xây dựng bot như thế nào.

II. Chi tiết các node trong thư viện

1. Telegram receiver

Telegram receiver node: Nút này nhận những tin nhắn từ Telegram gửi đến bot và cũng có thể nhận những tin nhắn từ một khung chat.



Cấu hình:

Edit receiver node

Delete Cancel Done

Properties [Settings] [Document] [Preview]

Bot Add new telegram bot... [Edit]

Name Name

Download Directory Download directory

Filter ☐ commands (from configured command nodes)

- Cấu hình bắt buộc : Bot
- Các mục cấu hình tự chọn là :
 - + Download Directory : Khi nút Telegram Receiver nhận các dữ liệu như là video, tệp tài liệu, v.v... thì các tệp tin sẽ được tải xuống tự động vào trong ổ đĩa cứng cục bộ khi thuộc tính này được cài đặt cấu hình. Đường dẫn thư mục cũng có thể là một phần của message payload : `msg.payload.path`. Ngoài ra, đối tượng tin nhắn có thể chứa liên kết tải xuống trực tiếp trong payload : `msg.payload.weblink`.
 - + Name : Thông thường, một nút Telegram Receiver nhận tất cả nội dung được gửi đến bot. Tuy nhiên, nếu bạn có các nút bên cạnh Telegram Receiver, bạn có thể bật cờ lệnh (commands flag) trong thuộc tính Filter để các lệnh dành cho một nút lệnh sẽ không được xử lý bởi nút Receiver.

Output:

- Tin nhắn thông thường từ thư viện nút bên dưới được lưu trữ trong `msg.originalMessage`. `Msg.payload` chứa dữ liệu quan trọng nhất như `chatId`, `type` và `content`. Các phần tử bổ sung có trong cấu trúc `msg.payload` và phụ thuộc vào loại tin nhắn.
- Nút có hai đầu ra:
 - + Đầu ra phía trên của nút (Đầu ra chuẩn) được sử dụng nếu thông báo đến từ một người dùng được ủy quyền.
 - + Đầu ra thấp hơn của nút (Đầu ra trái phép) được sử dụng khi áp dụng bảo mật (thông qua thuộc tính cấu hình Người dùng và `ChatIds`) và người dùng không được phép truy cập vào bot.

2. Telegram sender Node

Nút này gửi nội dung đến người dùng Telegram hoặc trò chuyện qua Telegram. Nó được kích hoạt bởi một đối tượng msg đến ở đầu vào của nó có chứa tất cả thông tin Telegram cần thiết.

Đối tượng msg đầu vào của nút tương tự như đối tượng msg đầu ra của nút nhận Telegram.

Cấu hình:

Các mục cấu hình bắt buộc là:

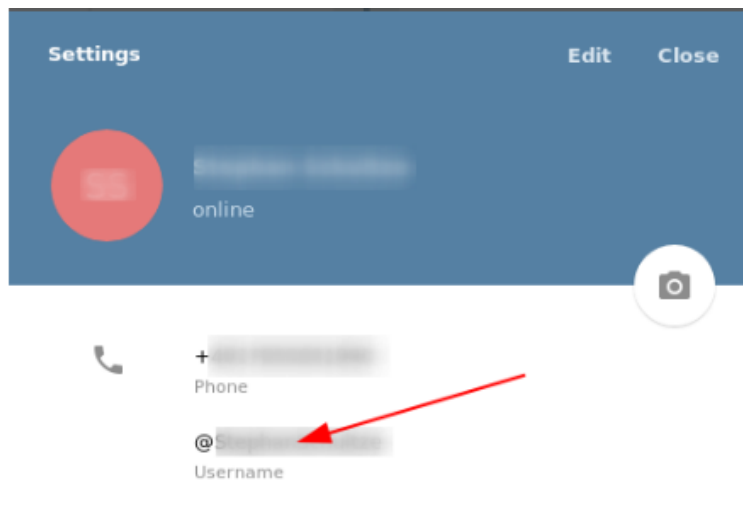
the Bot-Name and the Token

The screenshot shows the configuration window for the Telegram sender node. At the top right are 'Cancel' and 'Add' buttons. The 'Properties' tab is active. The 'Bot-Name' field is labeled '(Name of bot to connect to)' and the 'Token' field is labeled '(Enter the bot token from botfather here)'. Below these are optional fields for 'Users' (labeled '(Optional list of authorized user names e.g.: hugo,sepp,egon)'), 'ChatIds' (labeled '(Optional list of authorized chat-ids e.g.: -1234567,2345678,-34567)'), and 'Server URL' (labeled '(Optional URL for proxying and testing e.g.: https://api.telegram.org)'). The 'Update Mode' is set to 'Polling'. Under 'Polling Options', the 'Poll Interval' is set to 300. A tip box states: 'Tip: Polling mode is very robust and easy to set up. Nevertheless it creates more traffic on the network over time.' There are checkboxes for 'Use SOCKS5' and 'Verbose Logging'. Another tip box states: 'Tip: When verbose logging is turned on additional log messages will be printed to the node-red console.'

Các mục cấu hình tự chọn là:

Users and ChatIds.

Bạn có thể nhập danh sách các username, chatIDs được phép sử dụng bot này.



Hình 4: Hộp thoại cài đặt ứng dụng Telegram (ví dụ ứng dụng điện thoại Android)

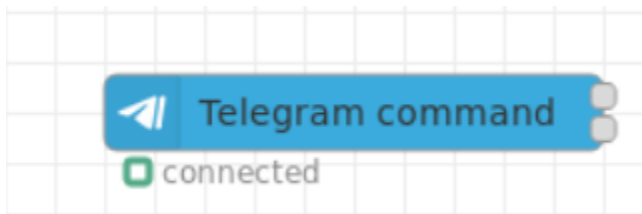
Đầu vào `msg.payload` phải chứa các phần tử sau:

`msg.payload.chatId` - chatId hoặc một loạt các chatId nếu bạn muốn gửi cùng một tin nhắn cho nhiều cuộc trò chuyện

`msg.payload.type` - ví dụ: tin nhắn , tài liệu , ảnh , v.v. (xem phần Nút người nhận ở trên)

`msg.payload.content` - nội dung tin nhắn của bạn (ví dụ: nội dung tin nhắn)

3. Command Node



Node này có thể được sử dụng để kích hoạt thông báo khi nhận được lệnh cụ thể: ví dụ `/help`.

Nếu bạn có nhiều bot trong một cuộc trò chuyện nhóm thực hiện cùng một lệnh, ví dụ

`/help`, bạn nên gửi lệnh trực tiếp đến một bot chuyên dụng bằng cách sử dụng ký hiệu `/help@`

YourBot đầy đủ để tránh việc các bot khác nhau sẽ được kích hoạt cùng một lúc.

Cấu hình:

Command: Lệnh sẽ được thực hiện.

Các lệnh thường bắt đầu bằng một dấu '/', ví dụ / help.

Register at telegram server: Để tự động đăng ký lệnh và mô tả tại máy chủ Telegram.

Strict in group chats: Để tránh việc bot xử lý các lệnh không được gửi trực tiếp đến nó bằng cách sử dụng ký hiệu dài (ví dụ /foo @ YourBot), bạn có thể đặt chế độ "strict" trong các tùy chọn của nút lệnh. Trong trường hợp này, bot chỉ chấp nhận ký hiệu lệnh đầy đủ trong các cuộc trò chuyện nhóm.

Has response output: Điều này cho phép đầu ra thứ hai. Nếu không thì nút chỉ có một đầu ra duy nhất.

Use Regex: Cho phép sử dụng biểu thức chính quy làm lệnh.

VD: ^ /toggle_ cho phép tất cả các lệnh bắt đầu bằng /toggle_

Remove command: loại bỏ lệnh khớp khỏi tin nhắn (sử dụng được khi Use Regex được chọn)

4. Event Node



Even Node kích hoạt đầu ra khi nhận được 1 sự kiện từ 1 cuộc trò chuyện.

Node nhận các sự kiện từ bot như:

Callback Query: trả về giá trị khi người dùng nhấp phím.

Inline Query: Trả về giá trị mặc định khi người dùng gửi giá trị null.

Edited Message: được kích hoạt khi ai đó thay đổi tin nhắn đã được gửi.

Edited Message Text: được kích hoạt khi ai đó thay đổi nội dung tin nhắn đã được gửi.

Edited Message Caption: được kích hoạt khi ai đó thay đổi chú thích của media đã được gửi

Channel Post: được kích hoạt khi bot là thành viên của kênh công khai

Edited Channel Post: được kích hoạt khi ai đó thay đổi tin nhắn đã được gửi trong kênh công khai.

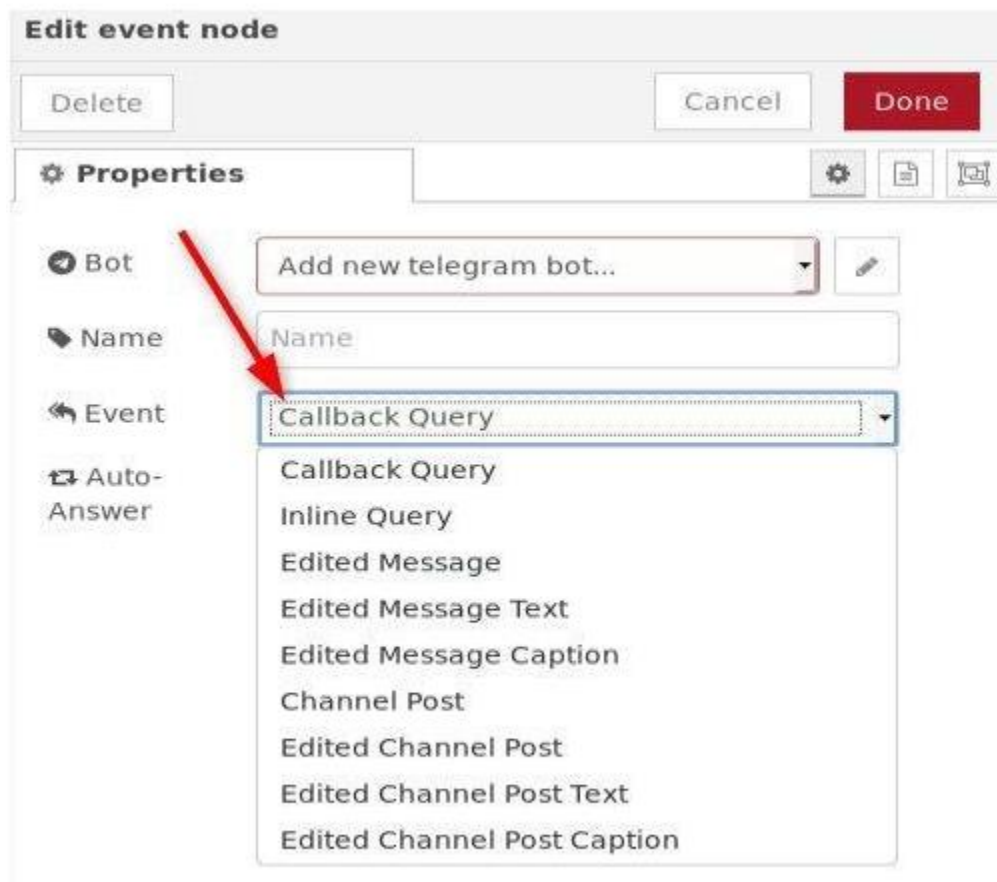
Edited Channel Post Text: được kích hoạt khi ai đó thay đổi nội dung tin nhắn đã được gửi trong kênh công khai.

Edited Channel Post Caption: được kích hoạt khi ai đó thay đổi chú thích đã được gửi về

Pre-Checkout Query: được kích hoạt khi ai đó phát hành thanh toán (xem gửi hóa đơn).

Shipping Query: được kích hoạt khi ai đó yêu cầu giao hàng.

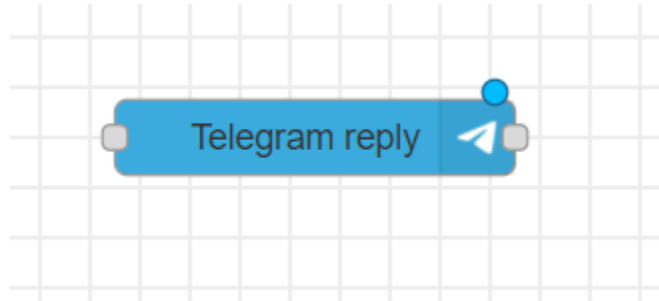
Cấu hình:



Auto-Answer: có thể được đặt cho Callback_Query. Nếu được kích hoạt, bạn không cần phải tự mình gửi câu trả lời rõ ràng cho bot.

5. Telegram Reply

Telegram Reply node : là một nút được kích hoạt khi ai đó trả lời một tin nhắn cụ thể. Nút Telegram Reply chờ câu trả lời cho một tin nhắn được chỉ định. Nó nên được sử dụng cùng với nút Sender.



Hình ảnh nút Telegram Reply

Cấu hình: Bắt buộc phải có là Bot

Edit reply node

Delete

Cancel

Done

⚙️ Properties

⚙️

📄

🔗

🤖 Bot

Test CLC4

▼

✎

🏷️ Name

Name

Input:

Đầu vào chuẩn: Gọi onReplyToMessage của bot.

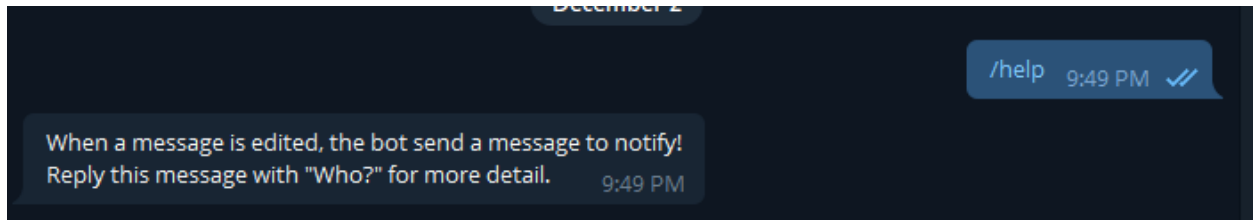
Output:

- Đầu ra chuẩn: Chứa kết quả từ việc gọi onReplyToMessage.
- Nút này có thể hữu ích trong vài trường hợp. Ví dụ: đã gửi một tin nhắn và bạn muốn thực hiện một số hành động khi ai đó trả lời tin nhắn được chỉ định này. Trả lời tin nhắn được thực hiện bằng cách nhấp vào tin nhắn trong ứng dụng khách của bạn và chọn câu trả lời từ cửa sổ bật lên.
- Msg.payload chứa:
 - + chatId: ChatId đích
 - + sentMessageId: Id của tin nhắn đã gửi trước đó trong cuộc trò chuyện
 - + content: Nội dung tin nhắn.

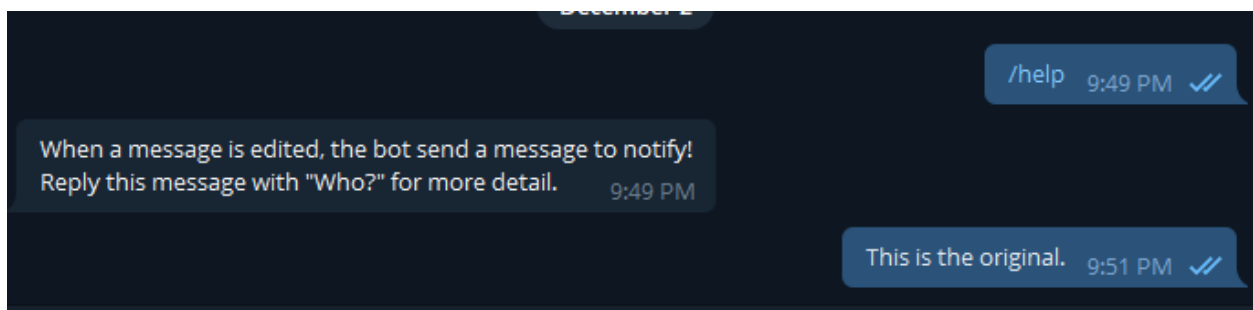
III. Kịch bản sử dụng thư viện và cài đặt:

1. Kịch bản:

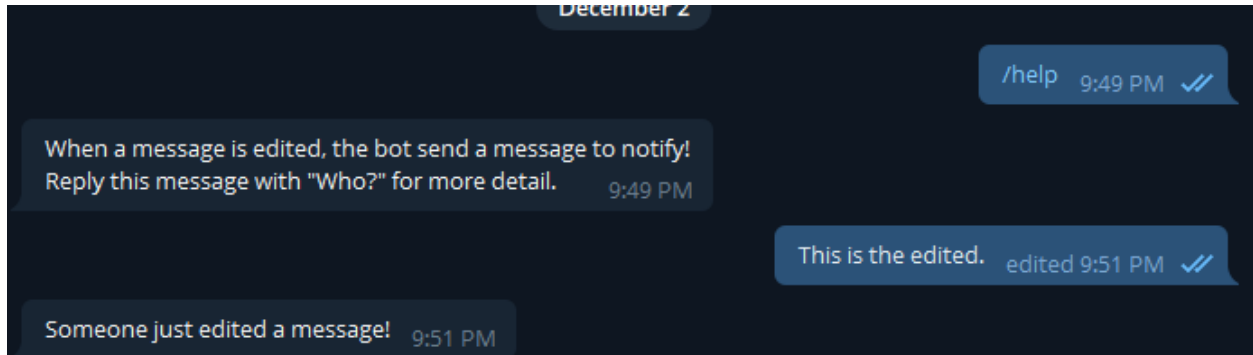
Người dùng gõ lệnh /help để tìm hiểu về lệnh của Bot, sau đó thực hiện theo.



Người dùng sẽ gửi một tin nhắn

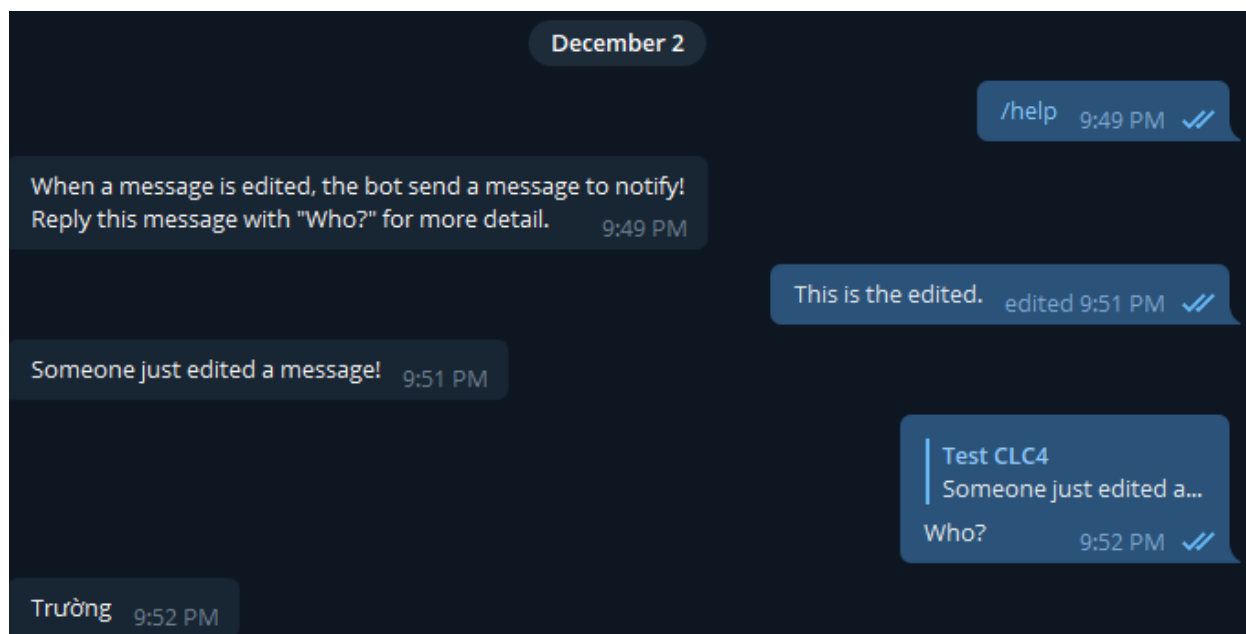


Và sau đó chỉnh sửa nó.

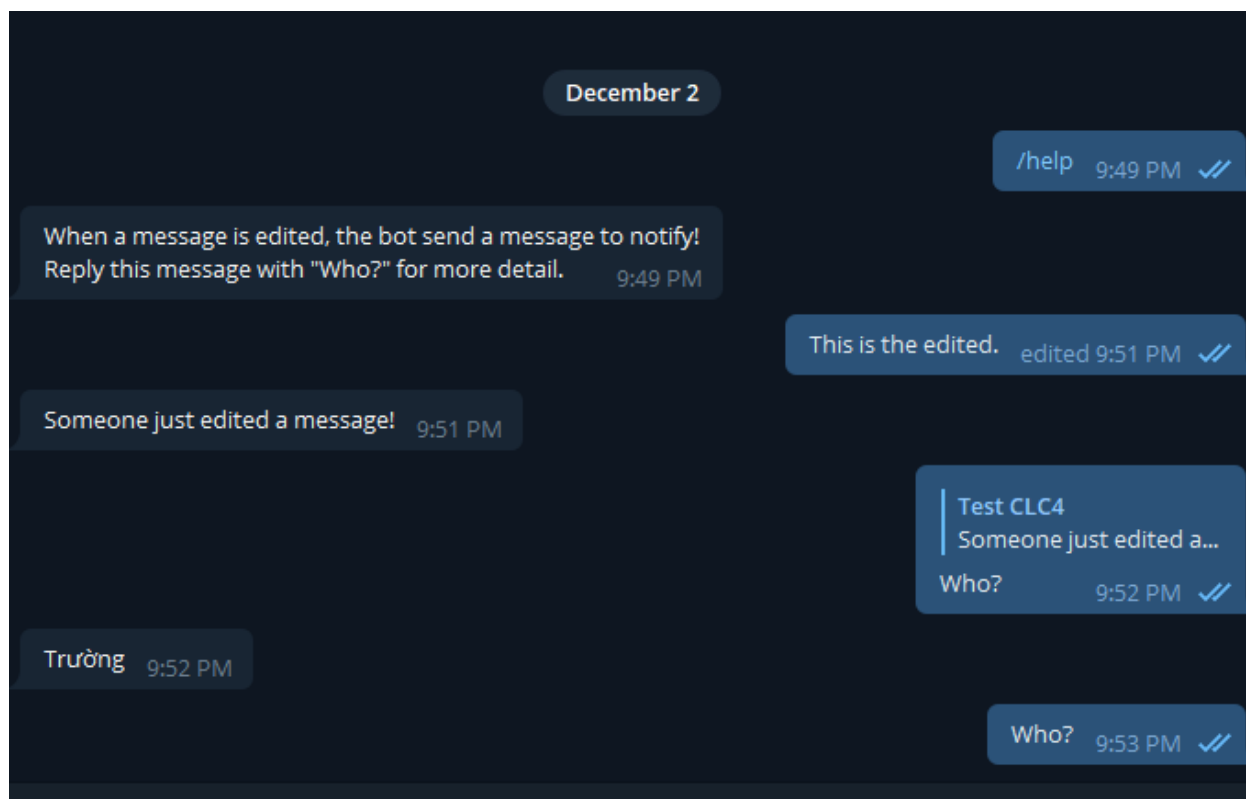


Bot sẽ thông báo có người chỉnh sửa tin nhắn.

Người dùng sẽ trả lời tin nhắn này với cú pháp 'Who?' để biết ai đã chỉnh sửa.

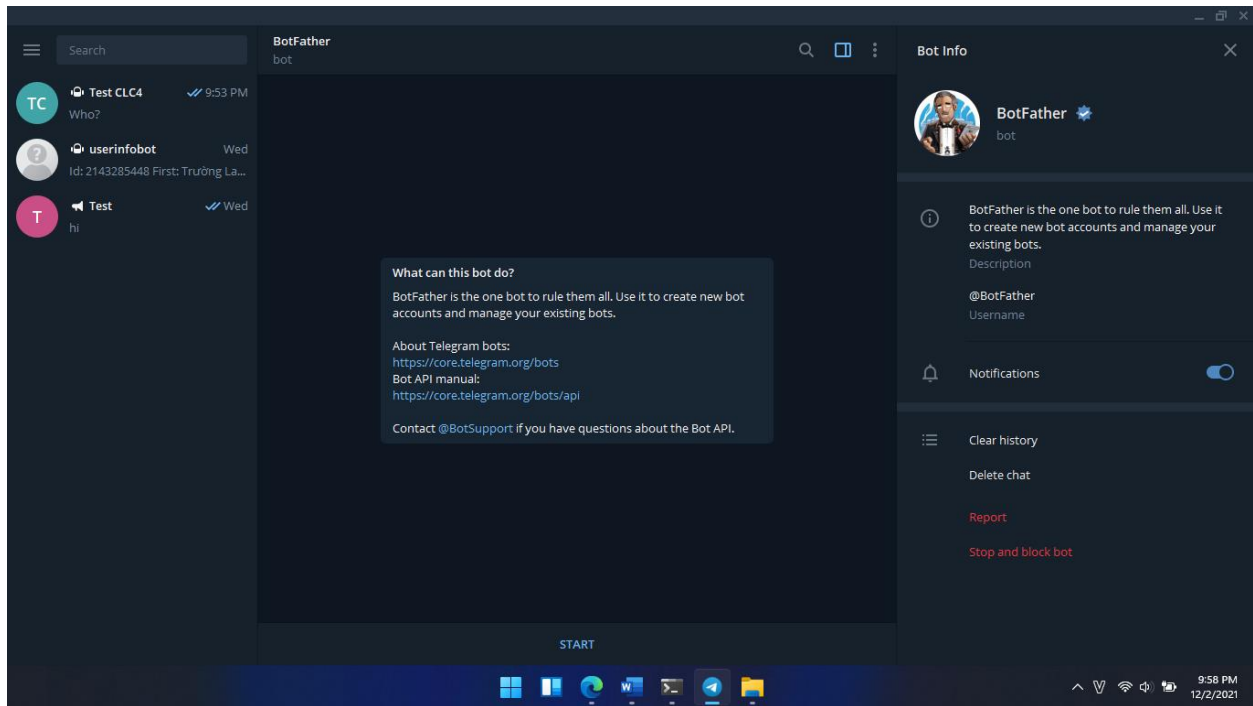


Chỉ khi người dùng trả lời (không phải gửi tin nhắn đơn thuần) thì Bot mới gửi thông tin người chỉnh sửa.



2. Cài đặt:

Truy cập Telegram và tìm BotFather (Đây là Bot của Telegram hỗ trợ người dùng tạo Bot)



Sau đó, nhấn bắt đầu, lúc này Bot sẽ gửi danh sách các lệnh có thể sử dụng cho người dùng:

BotFather

bot

I can help you create and manage Telegram bots. If you're new to the Bot API, please [see the manual](#).

You can control me by sending these commands:

`/newbot` - create a new bot
`/mybots` - edit your bots **[beta]**

Edit Bots

`/setname` - change a bot's name
`/setdescription` - change bot description
`/setabouttext` - change bot about info
`/setuserpic` - change bot profile photo
`/setcommands` - change the list of commands
`/deletebot` - delete a bot

Bot Settings

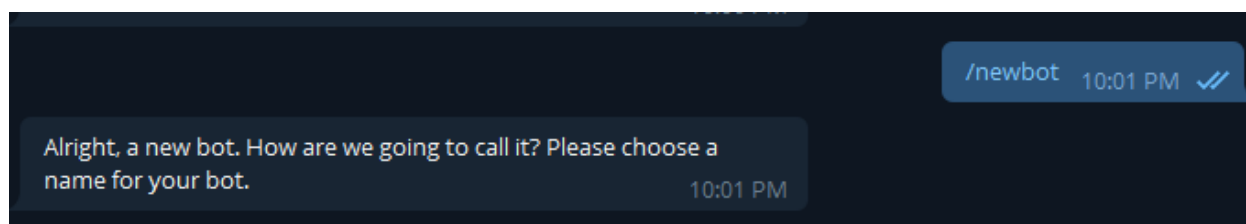
`/token` - generate authorization token
`/revoke` - revoke bot access token
`/setinline` - toggle inline mode
`/setinlinegeo` - toggle inline location requests
`/setinlinefeedback` - change inline feedback settings
`/setjoingroups` - can your bot be added to groups?
`/setprivacy` - toggle privacy mode in groups

Games

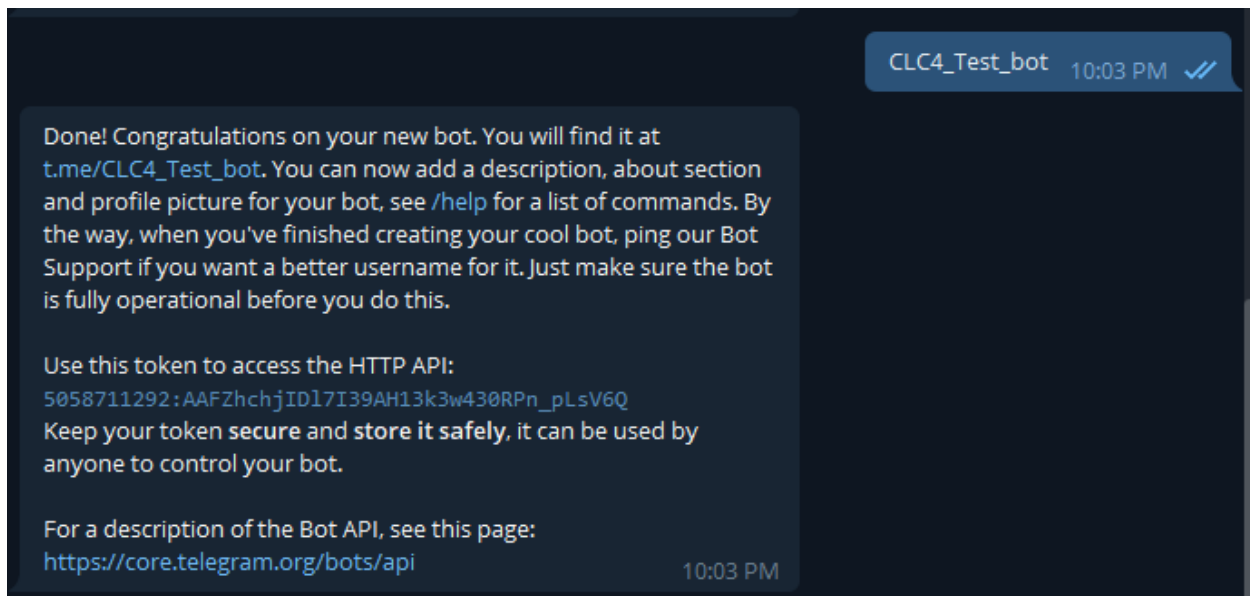
`/mygames` - edit your games **[beta]**
`/newgame` - create a new game
`/listgames` - get a list of your games
`/editgame` - edit a game
`/deletegame` - delete an existing game

10:00 PM

Gõ `/newbot` để tạo Bot mới cho bản thân:

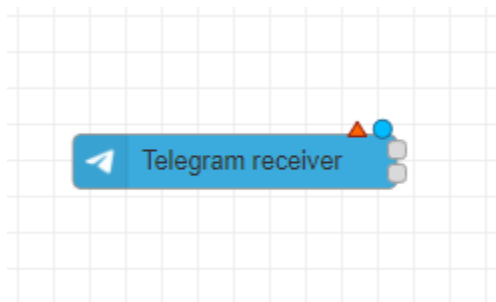


Nhập tên cho Bot của mình:

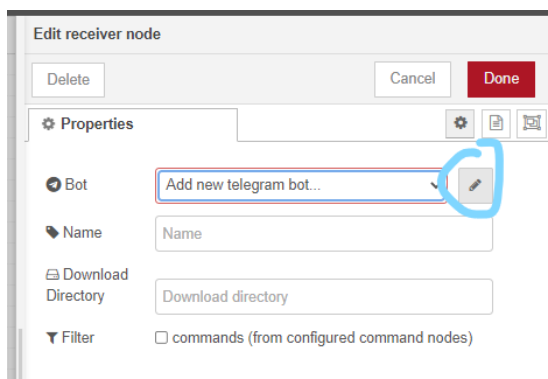


Sau khi Bot được tạo, ta sẽ được cấp HTTP API và đường dẫn đến Bot của mình.

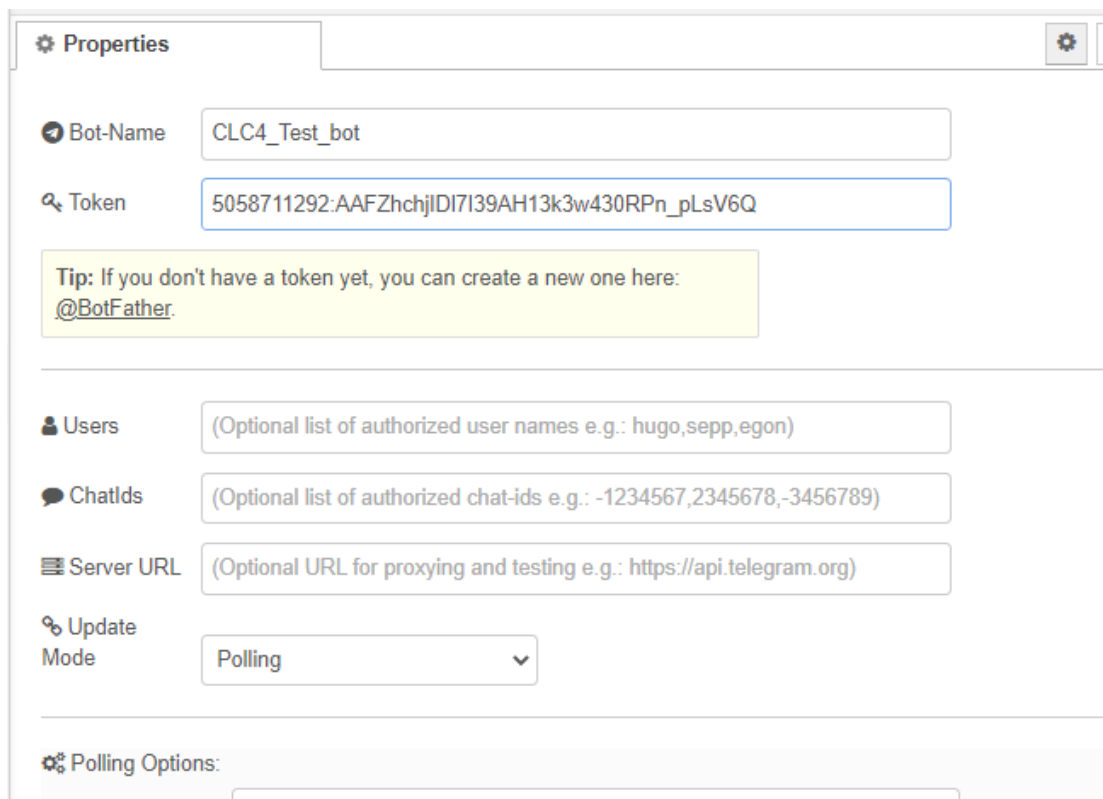
Tạo một node receiver:



Sau đó, nhấp đôi chuột vào node receiver để tạo một Bot mới (click nút edit)



Trong bảng Properties, nhập tên Bot và API của Bot (những dòng còn lại bỏ trống)



Properties

Bot-Name CLC4_Test_bot

Token 5058711292:AAFZhchjIDi7I39AH13k3w430RPn_pLsV6Q

Tip: If you don't have a token yet, you can create a new one here: [@BotFather](#).

Users (Optional list of authorized user names e.g.: hugo,sepp,egon)

ChatIds (Optional list of authorized chat-ids e.g.: -1234567,2345678,-3456789)

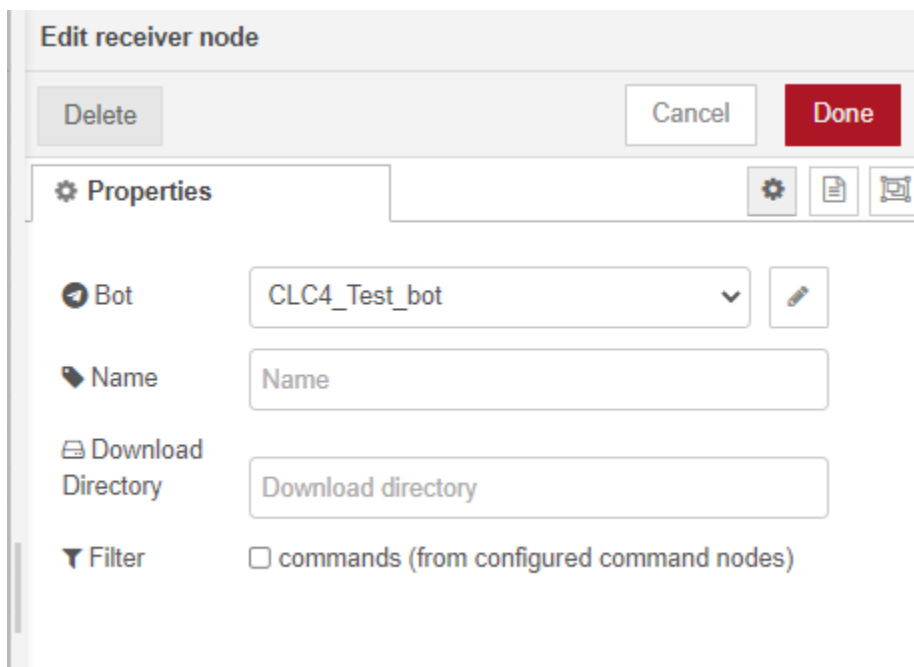
Server URL (Optional URL for proxying and testing e.g.: https://api.telegram.org)

Update Mode Polling

Polling Options:

Lúc này, ta có thể gán label (hoặc không) cho node.

Download Directory để trống.



Edit receiver node

Delete Cancel Done

Properties

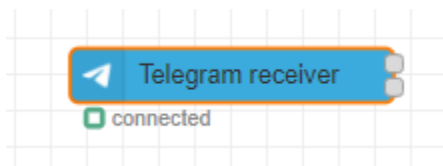
Bot CLC4_Test_bot

Name Name

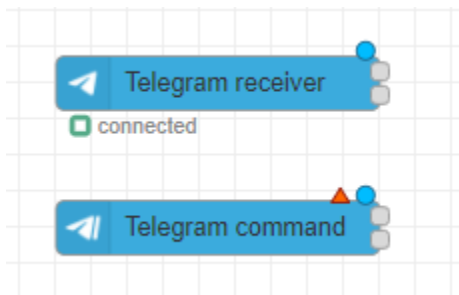
Download Directory Download directory

Filter ☐ commands (from configured command nodes)

Sau khi Deploy, Node báo Connected tức là đã hoàn thành kết nối với Bot

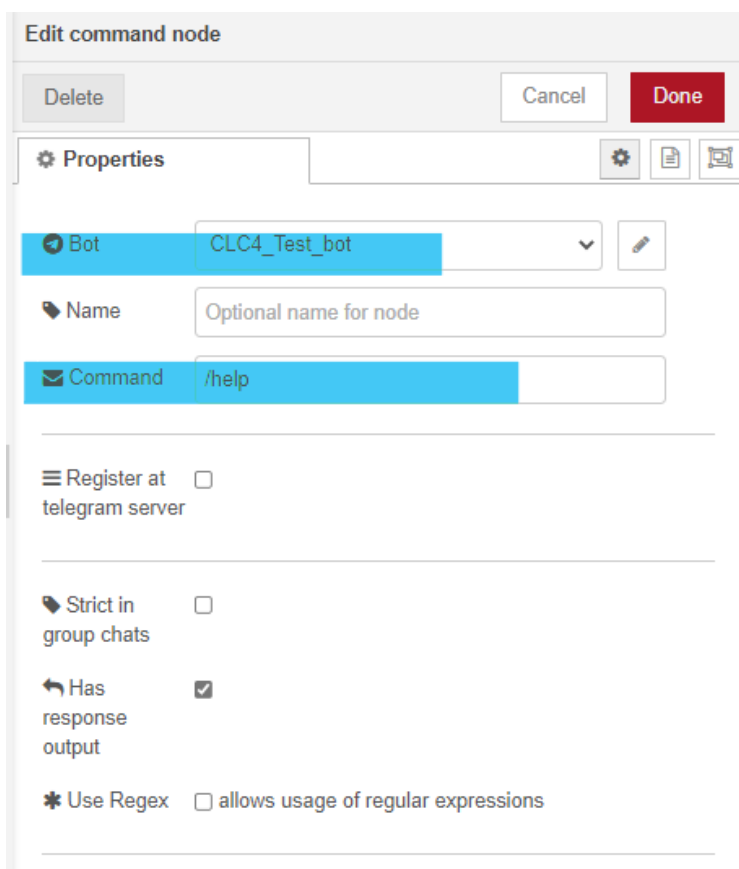


Để tạo một command, ta dùng node Command trong thư viện:



Cấu hình node, chọn Bot đã tạo và tên Command (ở đây là /help)

Những ô còn lại để mặc định.



Edit command node

Delete Cancel Done

Properties

Bot CLC4_Test_bot

Name Optional name for node

Command /help

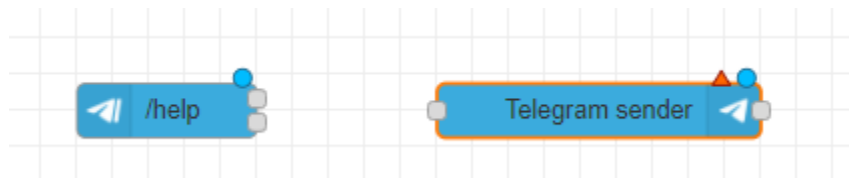
☐ Register at telegram server

☐ Strict in group chats

☒ Has response output

☐ Use Regex allows usage of regular expressions

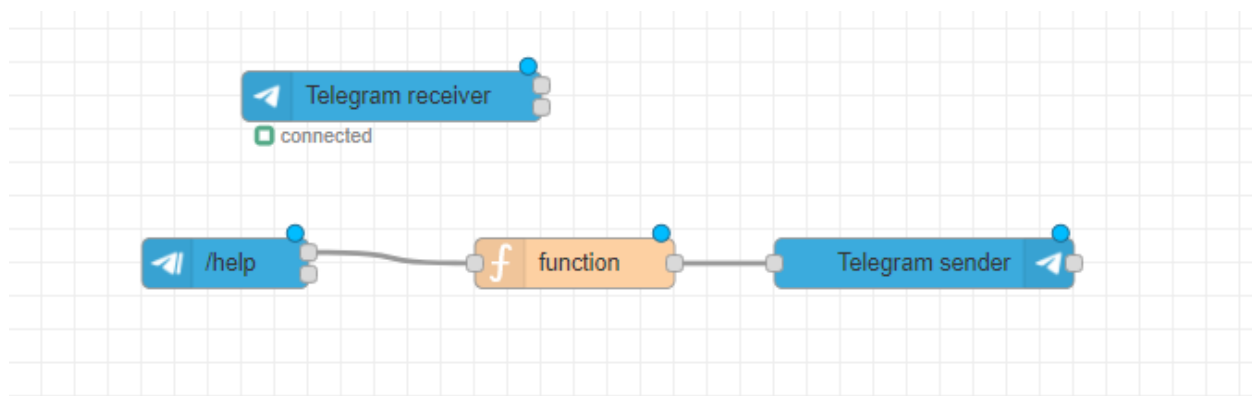
Sau khi người dùng gửi lệnh /help cho Bot, Bot sẽ gửi lại tin nhắn hỗ trợ cho người dùng thông qua node Sender:



Cấu hình node Sender

The screenshot shows the 'Properties' panel for a Telegram sender node. It includes a 'Bot' dropdown menu set to 'CLC4_Test_bot', a 'Name' text input field with the placeholder 'Name', and a checkbox for 'Send errors to second output' which is currently unchecked.

Để Bot có thể đọc command và gửi được tin nhắn, chỉ có 2 node này là chưa đủ, mà cần phải có thông tin tin nhắn Bot sẽ gửi. Lúc này ta cần node Function của Node-red để thiết lập.



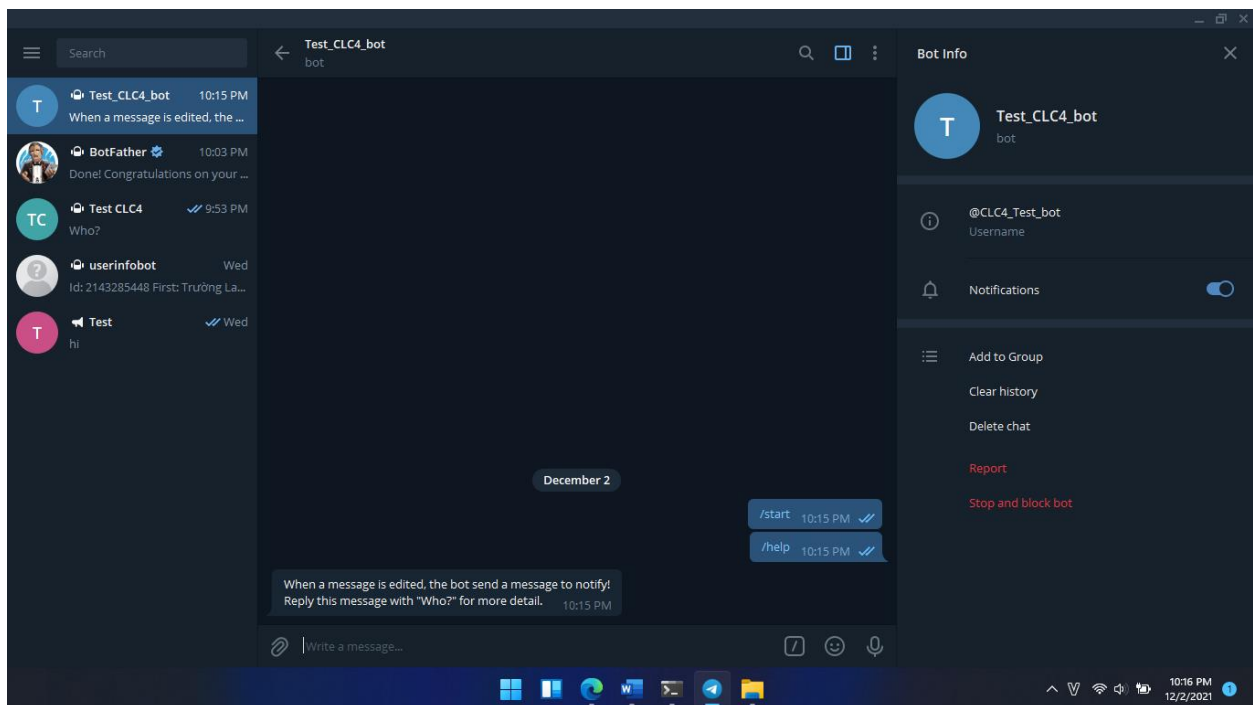
Cấu hình node function:

Ta sẽ đặt type của payload là message (tin nhắn văn bản thường)

Và content của message:



Sau khi Deploy, ta có thể gõ /help để thực hiện lệnh vừa tạo:

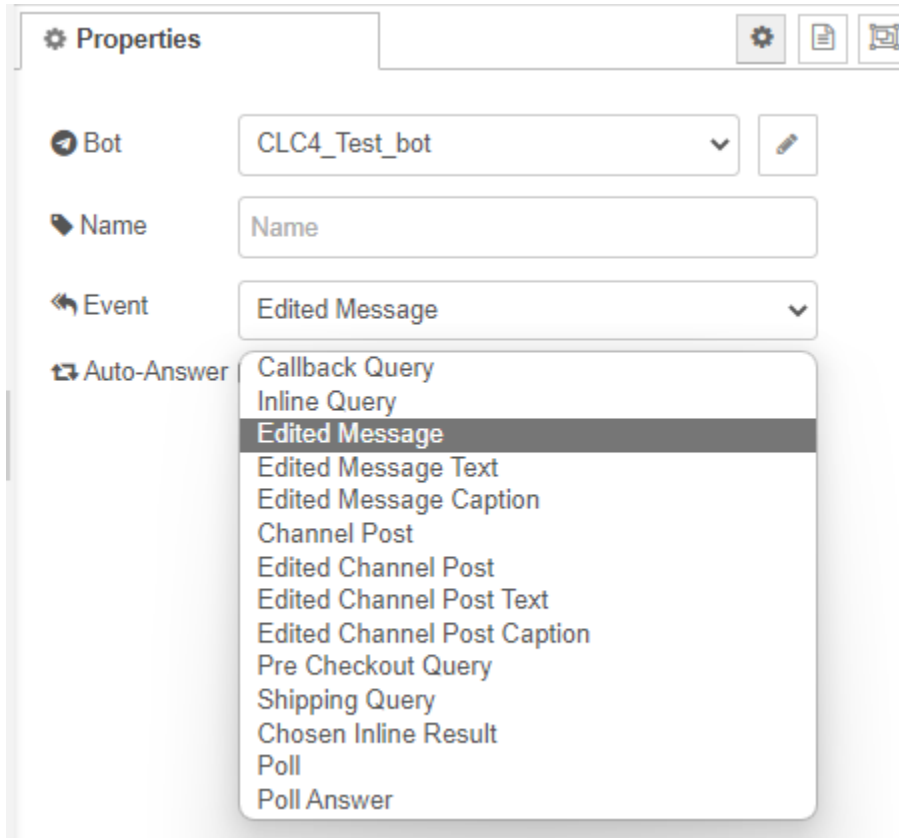


Ta sẽ cài đặt một kịch bản như đã nêu trong nội dung ở trên.

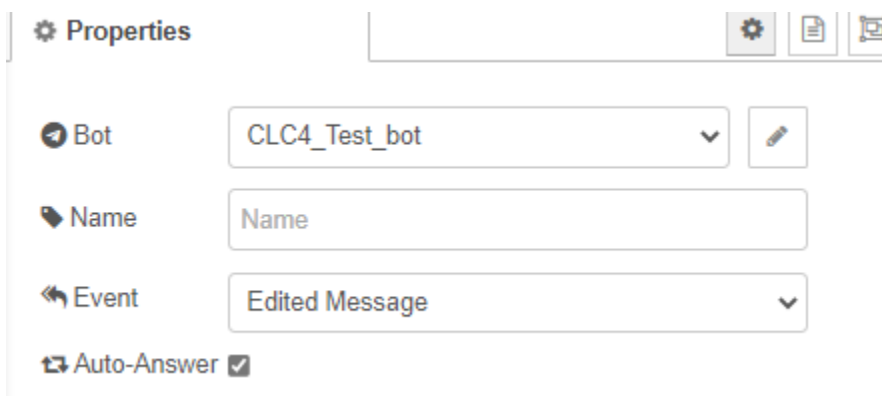
Để biết được ai vừa sửa tin nhắn, ta cần phải bắt được event đó. Việc này rất dễ dàng thông qua node event.



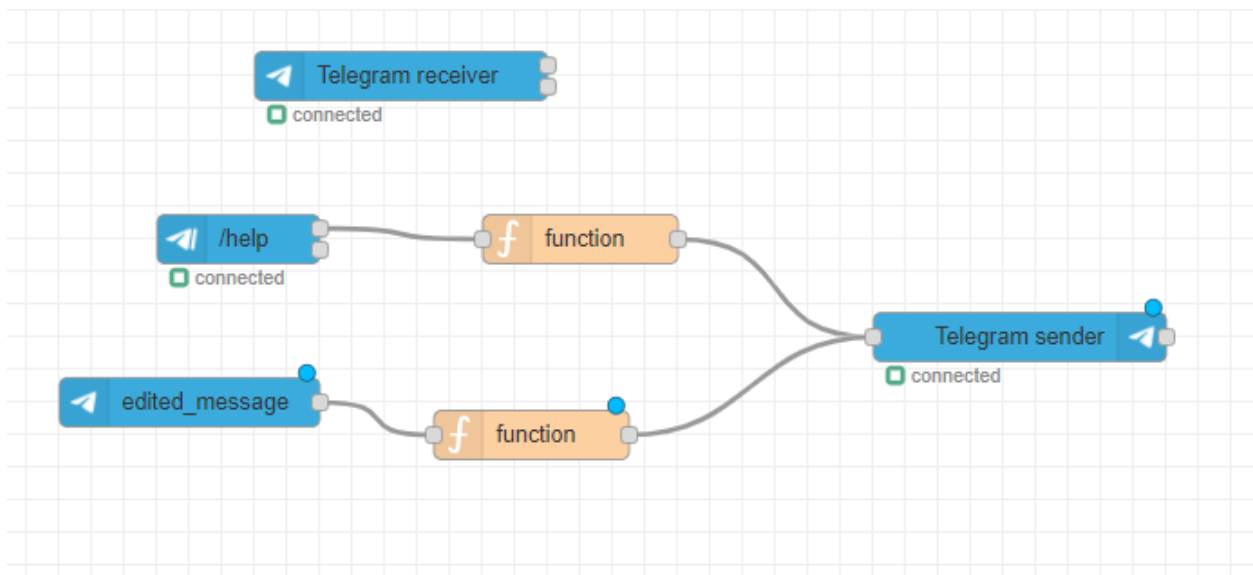
Thư viện này hỗ trợ nhiều event khác nhau, trong kịch bản này, chúng ta chọn Edited Message như trong hình dưới:



Để Bot có thể tự động thông báo khi có event xảy ra, ta chọn thêm mục Auto-Answer:



Bot sẽ gửi tin nhắn thông báo khi có người chỉnh sửa tin nhắn:



Cấu hình node function:

Ở đây, ta lưu thêm biến name để dành cho kịch bản sắp tới.

Edit function node

Delete

Properties

Name

Name

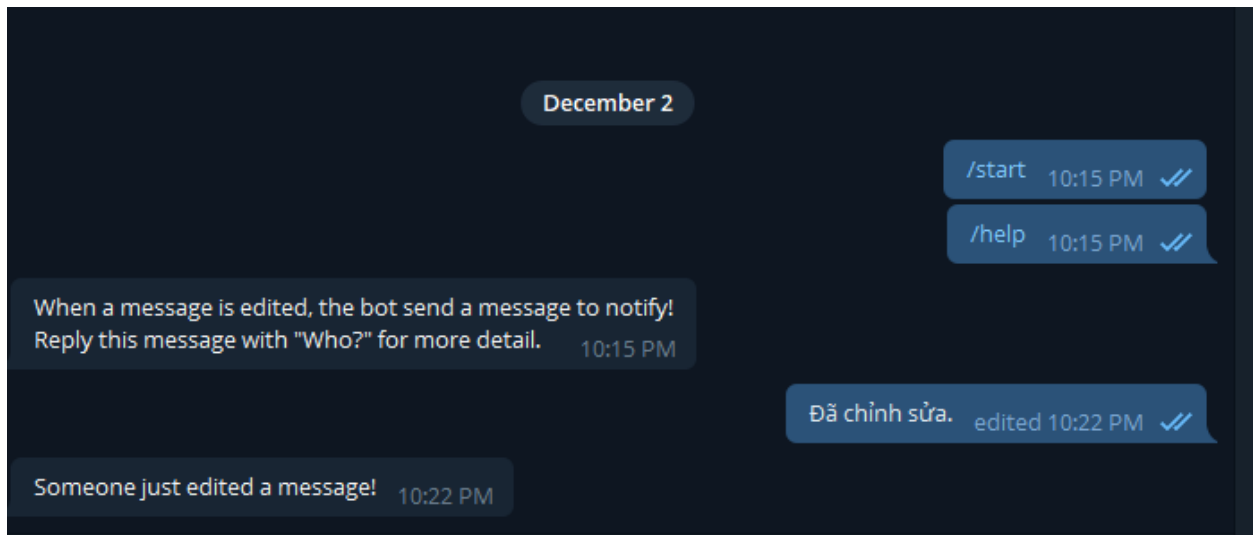
Setup

On Start

On Message

```
1 msg.payload.type = 'message'
2 msg.payload.content = 'Someone just edited a message!'
3 global.set('name',msg.payload.chat.first_name)
4 return msg;
```

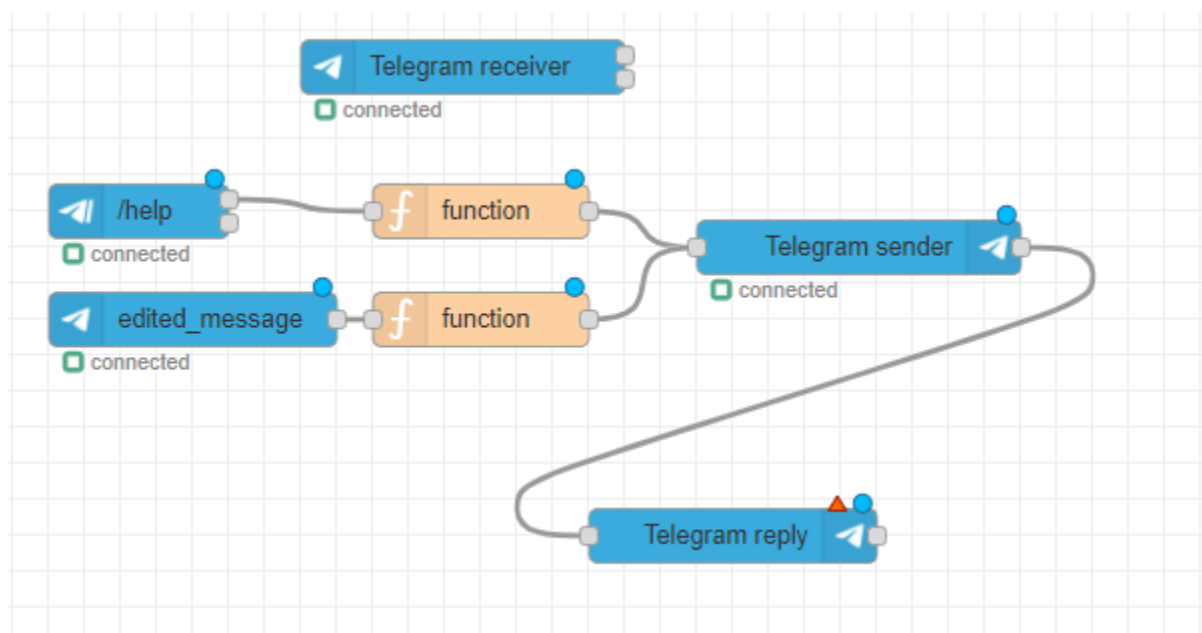
Ta thử chỉnh sửa một tin nhắn đã gửi:



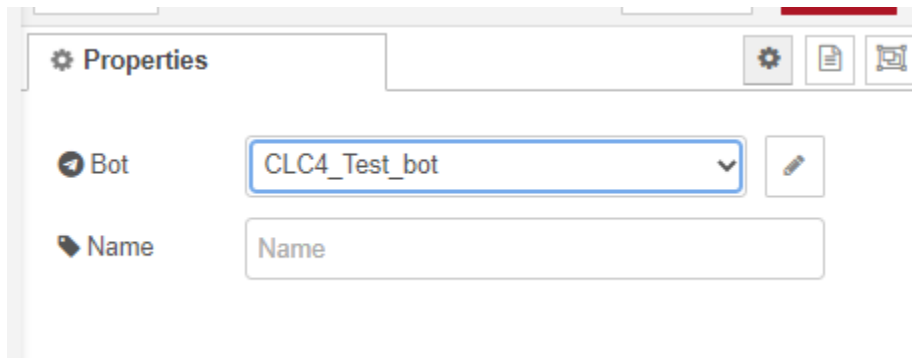
Khi người dùng muốn biết ai đã chỉnh sửa tin nhắn, người dùng cần trả lời tin nhắn của Bot với cú pháp 'Who?' (Theo kịch bản)

Để bắt được event có một câu trả lời của một tin nhắn, ta dùng node Reply:

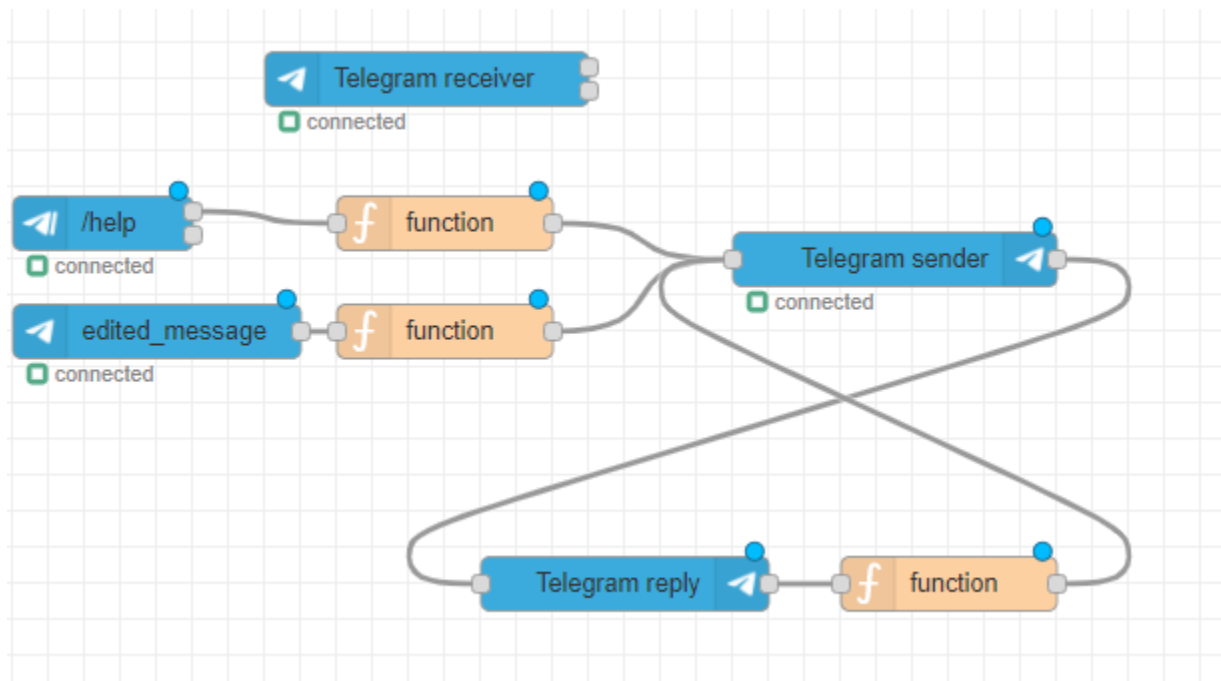
Ở đây ta sẽ bắt event cho tin nhắn của Bot:



Cấu hình node Reply:

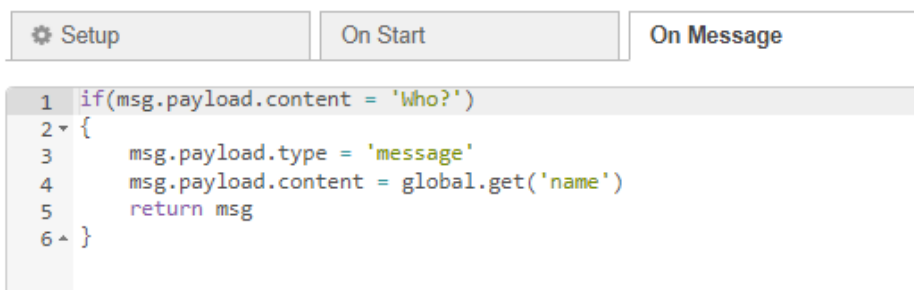


Nhưng đây chỉ là cấu hình khi có bất cứ tin nhắn trả lời nào được gửi đi, cho nên ta cần một node function để kiểm tra xem câu trả lời có đúng cú pháp không, sau đó sẽ trả về tên của người chỉnh sửa tin nhắn:



Ta sẽ tận dụng luôn node Sender.

Cấu hình node function:



Vậy là kịch bản đã hoàn tất, ta có thể Deploy và thử nghiệm:

