

LINE Chatbot with Dialogflow, Google Sheet and App Script



20 March 2023

shared slide: shorturl.at/erNS0

Asst. Prof. Dr. Aurawan Imsombut
Department of Computer Science, Kasetsart University

Agenda

- Chatbot
- LINE Chatbot
- Dialogflow
- Google Sheet and Google Action Script

Chatbots

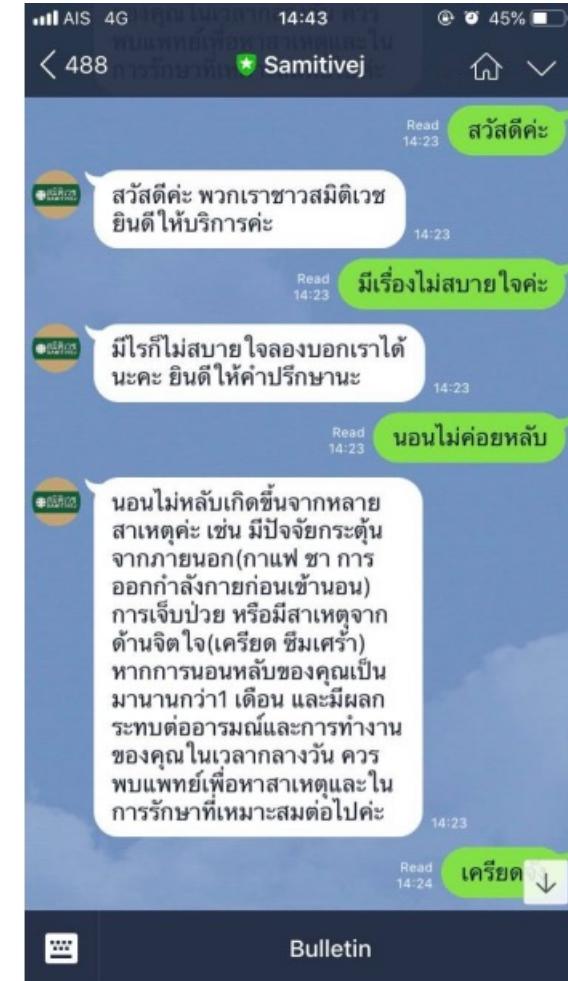
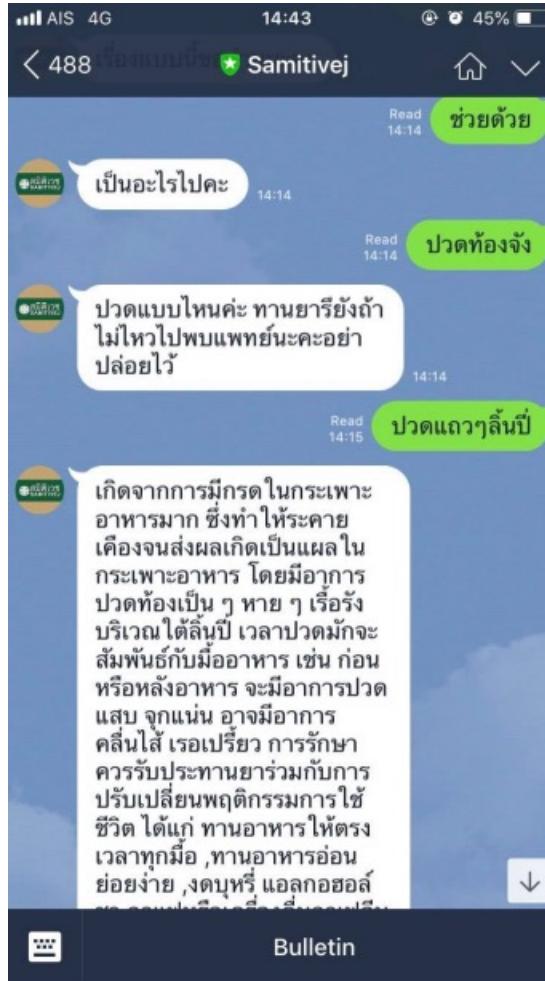
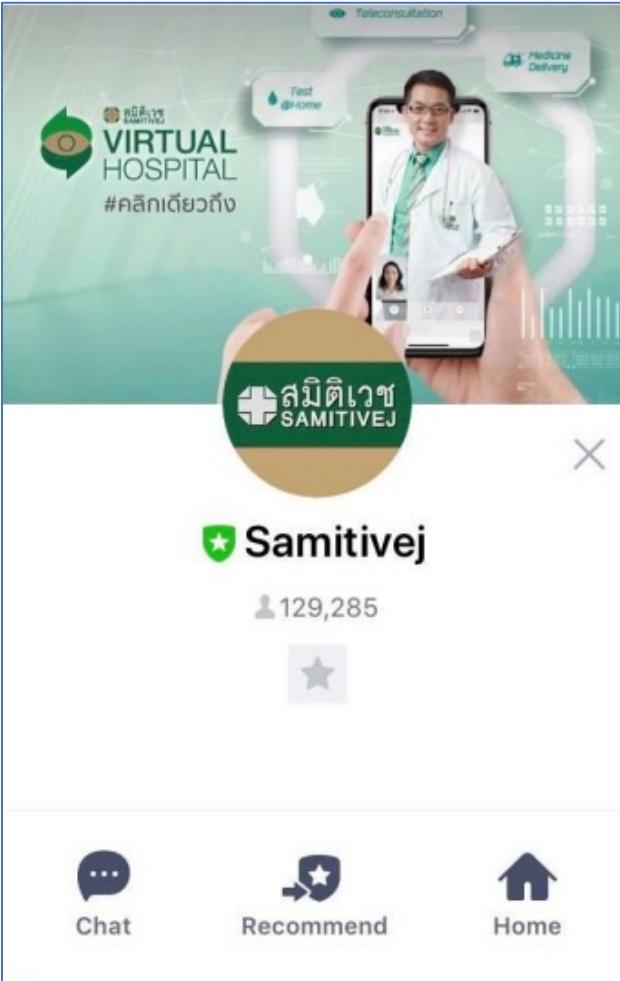
- Chatbots are essentially computer programs that are built to simulate human conversation.
- They are used in a variety of industries and settings, including *customer service*, *healthcare*, and *e-commerce*.
- They can handle a wide range of tasks, from answering basic questions to helping customers make purchases and even providing emotional support.
- They often run inside a popular messaging application such as Line, Facebook Messenger, or SMS.



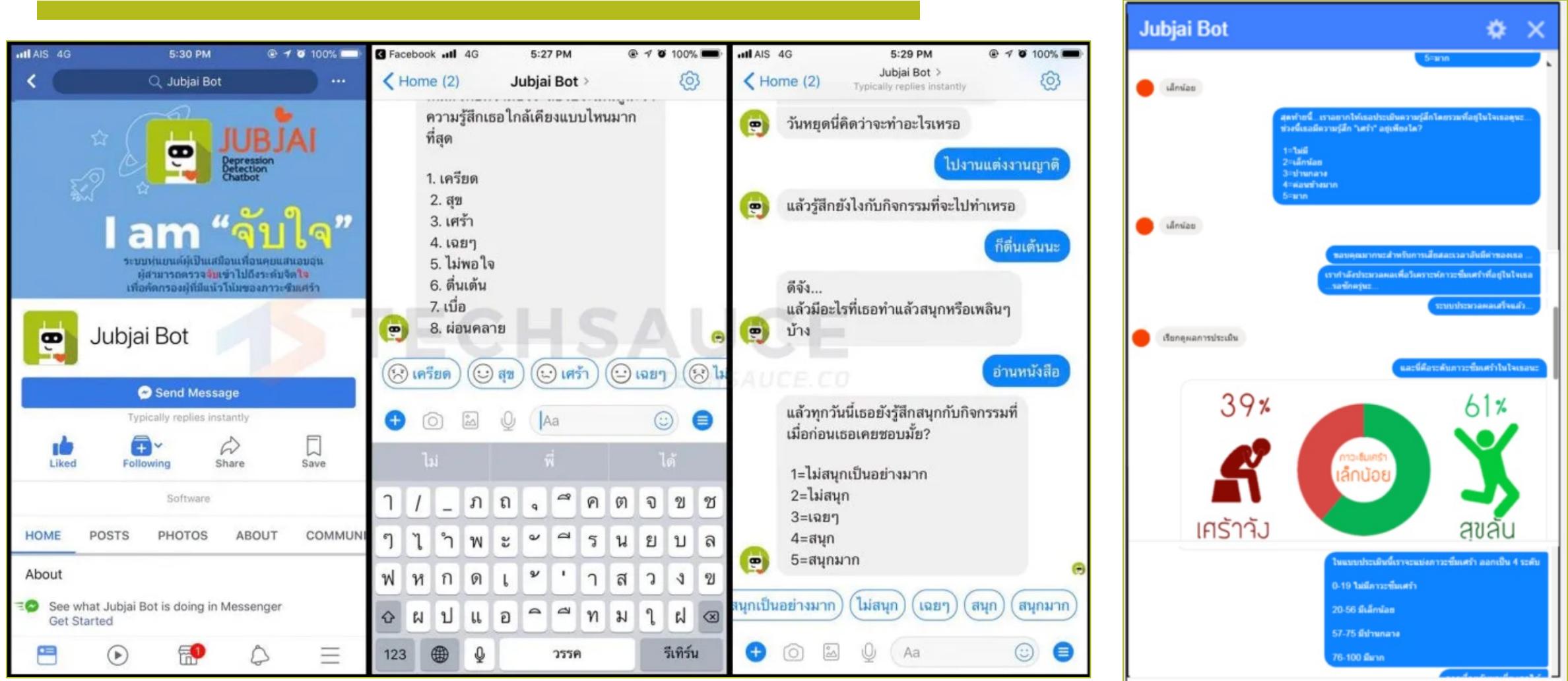
Chatbots

- The Two Types of Chatbots
 - **Rules Based chatbots**
 - It can only respond to very specific commands. If the user don't use the right command or words, the chatbot doesn't know what the meanings.
 - E.g., Google Dialogflow
 - **AI-powered chatbots**
 - Uses machine learning and AI to provide the best response
 - Understands language, as well as commands.
 - Can chat in a similar way a staff member would with a person.
 - E.g., chatGPT

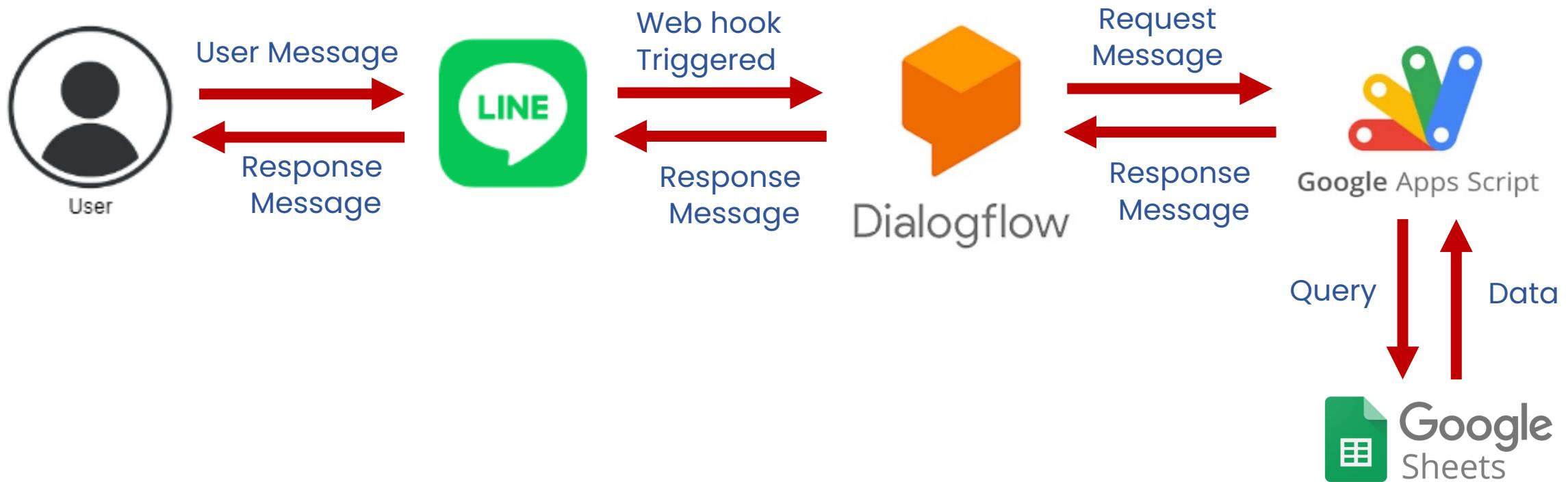
Samitivej Hospital Chatbot



Jubjai Bot: Depression Detection Chatbot



LINE Chatbot with Dialogflow and Google Sheet

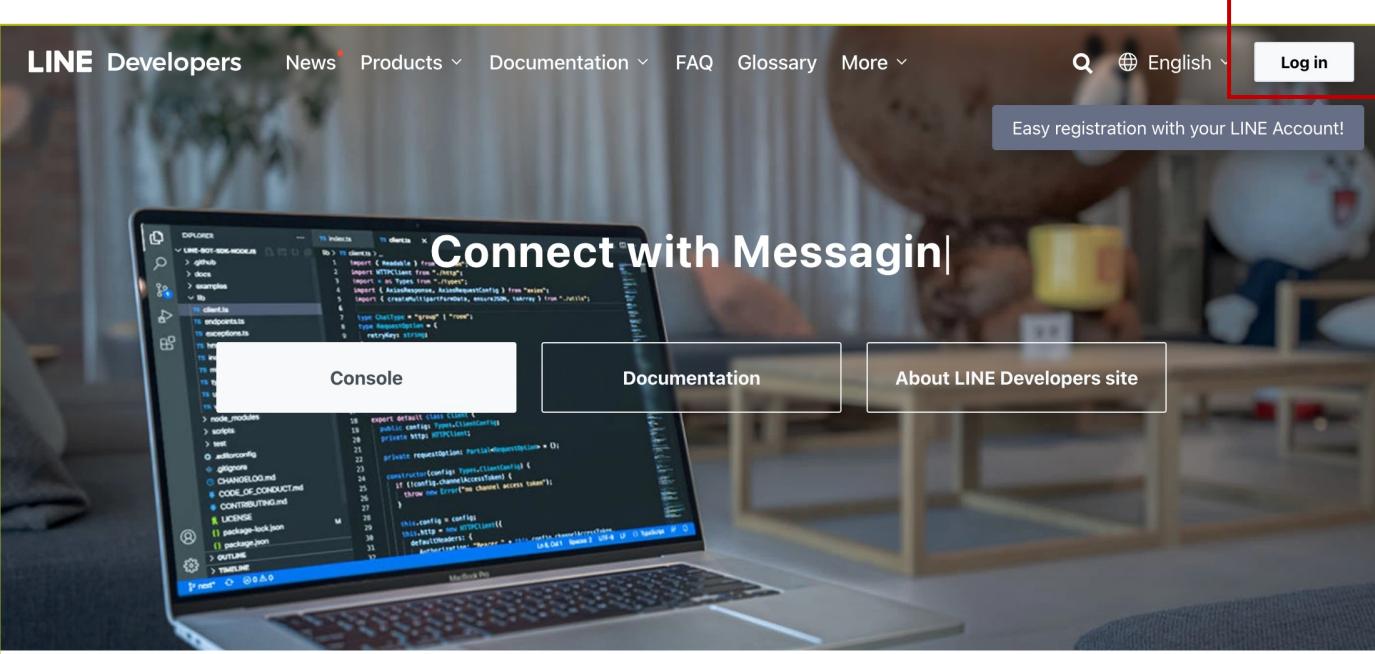




LINE Channel

Create LINE Channel

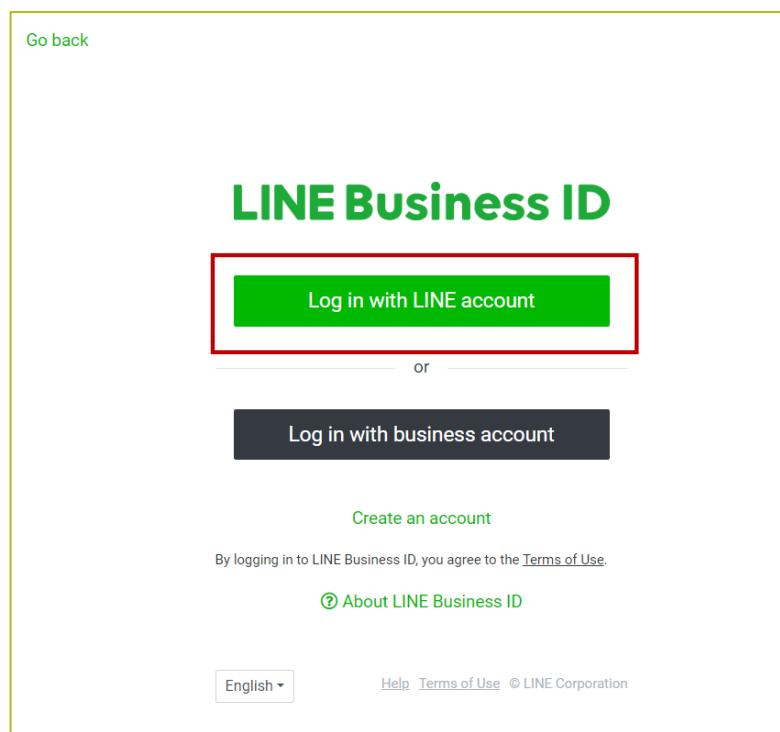
- Go to website: <https://developers.line.biz/console/>



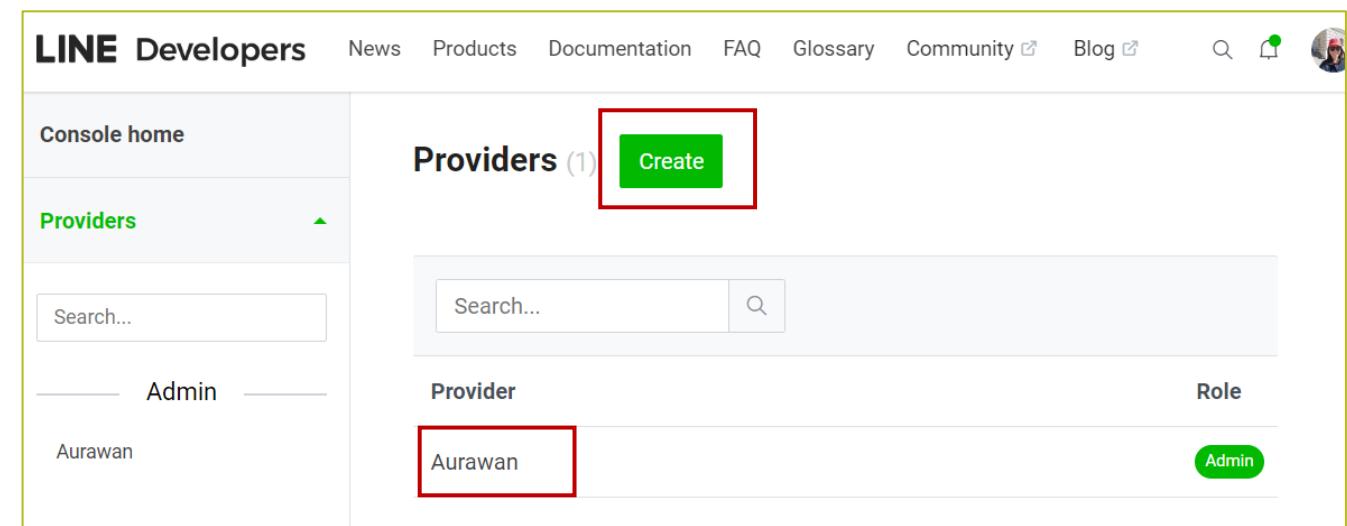
The screenshot shows the homepage of the LINE Developers site. At the top right, there is a "Log in" button with a red box drawn around it. Below the header, a banner says "Easy registration with your LINE Account!". In the center, there's a large image of a laptop displaying a code editor with some JavaScript code. Below the image are three buttons: "Console", "Documentation", and "About LINE Developers site". The "Console" button is currently active. At the bottom of the page, there's a section titled "Products" with a button that says "Check out use cases and app demos at LINE API Use Case!" with a small icon next to it. At the very bottom, there are links for "LINE Login" and "Messaging API".

Create LINE Channel

1. Login with LINE account



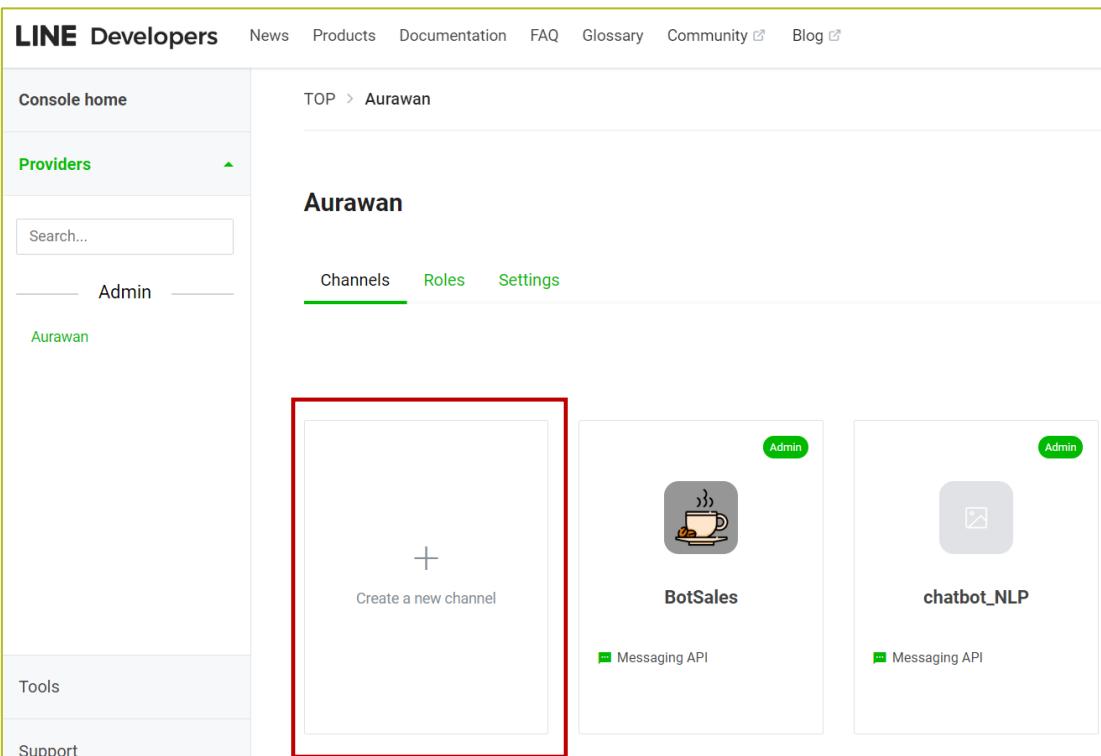
2. At the Provider section, click the Create button
Enter your provider's name and press the Create button.



The screenshot shows the 'LINE Developers' console. The top navigation bar includes links for News, Products, Documentation, FAQ, Glossary, Community, Blog, and user profile icons. The main area has a sidebar with 'Console home' and a 'Providers' section. Under 'Providers', there is a sub-section titled 'Provider' with a table. The table has columns for 'Provider' (containing 'Aurawan') and 'Role' (containing 'Admin'). A red box highlights the 'Create' button next to the 'Providers' count '(1)'. Another red box highlights the 'Aurawan' entry in the provider list.

Create LINE Channel

3. Click at your created Provider
and click "Create a new channel"



LINE Developers News Products Documentation FAQ Glossary Community Blog

Console home Providers Aurawan

Search... Admin

Aurawan

Channels Roles Settings

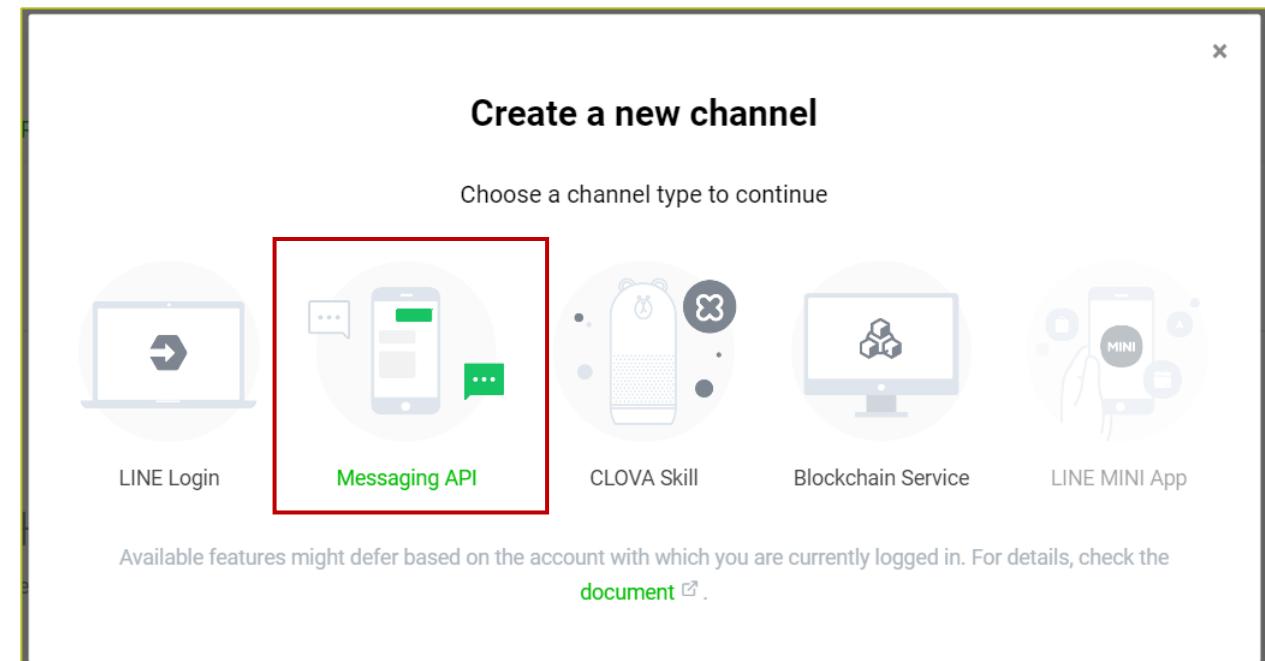
Create a new channel

BotSales Admin Messaging API

chatbot_NLP Admin Messaging API

Tools Support

4. Select Messaging API



Create a new channel

Choose a channel type to continue

LINE Login Messaging API CLOVA Skill Blockchain Service LINE MINI App

Available features might defer based on the account with which you are currently logged in. For details, check the document.

Create LINE Channel

5. Fill out all required information and press the Create button

TOP

Create a new channel

Channel type ▼
✓ Don't leave this empty

Provider ▼
✓ Don't leave this empty

Company or owner's country or region ▼
Corporations should select their company's country or region. Individuals should select the country or region of their store, or residence.
✓ Don't leave this empty

Channel icon optional 

TOP

Channel name G
Note: The channel name can't be changed for seven days.
✓ Don't leave this empty
✓ Don't use special characters (4-byte Unicode)
✓ Enter no more than 20 characters

Channel description G
✓ Don't leave this empty
✓ Don't use special characters (4-byte Unicode)
✓ Enter no more than 500 characters

Category ▼
✓ Don't leave this empty

Subcategory ▼
✓ Don't leave this empty

TOP

✓ Enter no more than 100 characters

Privacy policy URL optional
✓ Enter a valid HTTPS URL
✓ Enter no more than 500 characters

Terms of use URL optional
✓ Enter a valid HTTPS URL
✓ Enter no more than 500 characters

✓ I have read and agree to the [LINE Official Account Terms of Use](#) ↗
✓ I have read and agree to the [LINE Official Account API Terms of Use](#) ↗
✓ I have read and acknowledge [LINE Privacy Policy](#) ↗

✓ Select the checkbox after reading the related document

Create

Create LINE Channel

6. Click OK button

Create a Messaging API channel with the following details?

Channel name : First Aid

Official Account name : First Aid

Provider : Aurawan

- If you proceed, an official account will be created with the same name as the messaging API channel above.
- You cannot change the channel provider after the channel is created. Make sure that the provider and official account owner are the same individual developer, company or organization.
- For the handling of LINE user information, please refer to [User Data Policy](#).

Cancel

OK

7. Click Agree button

ข้อตกลงเกี่ยวกับการใช้ข้อมูล

LINE คอร์ปอเรชั่น ("LINE") จะมีการใช้ข้อมูลประเภทต่างๆ จากบัญชีของธุรกิจ ซึ่งรวมถึงแต่ไม่จำกัดเฉพาะบัญชีทางการ LINE (เรียกว่า "บัญชีทางการ") เพื่อพัฒนาและปรับปรุงการให้บริการ กรุณาตรวจสอบและยอมรับต่อข้อตกลงการใช้ข้อมูลก่อนใช้บัญชีทางการ

■ ข้อมูลที่จะเก็บและนำไปใช้

- เนื้อหาต่างๆ ที่มีการส่ง ทั้งที่ส่งถึงผู้ใช้และส่งจากผู้ใช้ เช่น ข้อความ ข้อมูล URL รูปภาพ วิดีโอ สติกเกอร์ และไฟล์ฟิล์ม เป็นต้น
- รูปแบบ ตัวเลข เวลาในการส่งหรือโทร คุณสนทนา และอื่นๆ ("รูปแบบและอื่นๆ") ของเนื้อหาทั้งหมดที่ส่งถึงหรือส่งจากผู้ใช้ รวมถึงรูปแบบและอื่นๆ ของเนื้อหาที่มีการสัมภาระใน VoIP (การโทรศัพท์อินเทอร์เน็ตและการโทรศัพท์วิธีอื่น)
- ที่อยู่ไอพีที่ใช้สำหรับบัญชีทางการ เวลาที่ใช้ในแต่ละไฟล์กัน สถานะว่าเนื้อหาถูกอ่านแล้วหรือไม่ รวมถึงการแตะหรือคลิก URL และอื่นๆ (รวมถึงแต่ไม่จำกัดเฉพาะลิงก์ที่เชื่อมไปยังแหล่งข้อมูล) ประวัติการใช้บริการ เช่น ประวัติการดูวีดีโอเบราว์เซอร์ผ่าน LINE และระยะเวลาที่ถูก ตลอดจนข้อมูลอื่นๆ ที่ระบุไว้ในนโยบายความเป็นส่วนตัว

■ วัสดุประสงค์ในการเก็บข้อมูล การใช้ และการส่งต่อให้กับบุคคลภายนอก

ข้อมูลดังต้นจะถูกนำไปใช้สำหรับ (1) การป้องกันการใช้งานโดยไม่ได้รับอนุญาต, (2) การให้บริการตลอดจนการพัฒนาและปรับปรุงบริการต่างๆ ของ LINE และ (3) การแสดงโฆษณาต่างๆ

นอกจากนี้ข้อมูลดังกล่าวอาจมีการส่งต่อให้กับบริษัทต่างๆ ในเครือของ LINE, บริษัทผู้ให้บริการ และบริษัทที่รับจ้างช่วงของ LINE

ข้อตกลงดังกล่าวมีไว้สำหรับผู้มีอำนาจในการใช้งานบัญชีทางการเท่านั้น หากเป็นบุคคลอื่น จำเป็นต้องได้รับความเห็นชอบจากผู้มีอำนาจ มีเขียนนั้น LINE อาจระงับการใช้งานบัญชีทางการหากได้รับแจ้งจากผู้มีอำนาจว่าไม่ได้รับการเห็นชอบโดย LINE จะไม่มีส่วนเกี่ยวข้องหรือรับผิดชอบกับผลที่เกิดขึ้นในภายหลัง

Agree

Create LINE Channel

- You will get a LINE Channel.

The screenshot shows the LINE Developers console interface. On the left, there's a sidebar with 'Console home', 'Providers' (selected), 'Search...', 'Admin' (underlined), 'Aurawan', 'Tools', and 'Support'. The main area shows 'TOP > Aurawan > First Aid'. The 'First Aid' channel is displayed with a blue first aid kit icon. It has tabs for 'Basic settings' (selected), 'Messaging API', 'LIFF', 'Security', 'Statistics', and 'Roles'. Under 'Basic settings', there's a section for 'Basic information' with a note: 'You can change your app name and icon in LINE Official Account Manager'. Below that are fields for 'Channel ID' (redacted) and 'Channel icon', which shows the same blue first aid kit icon.

Data for used LINE with Dialogflow

8. Click Basic Setting Tab, the **Channel ID** and **Channel Secret** are used in further work.

TOP > Aurawan > First Aid

First Aid
Admin | Messaging API

Basic settings (highlighted with a red box)
Messaging API LIFF Security Statistics Roles

Basic settings

Basic information

You can change your app name and icon in [LINE Official Account Manager](#).

Channel ID (highlighted with a red box) (1)

Channel icon

TOP > Aurawan > First Aid

App types Bot

Permissions ? PROFILE

Channel secret ? (2) **Issue**

Assertion Signing Key ?

Your user ID ? U8c6b15683b471c56d851630b182d5254

Delete this channel

Delete

Data for used LINE with Dialogflow



9. Next, click on the Messaging API tab.

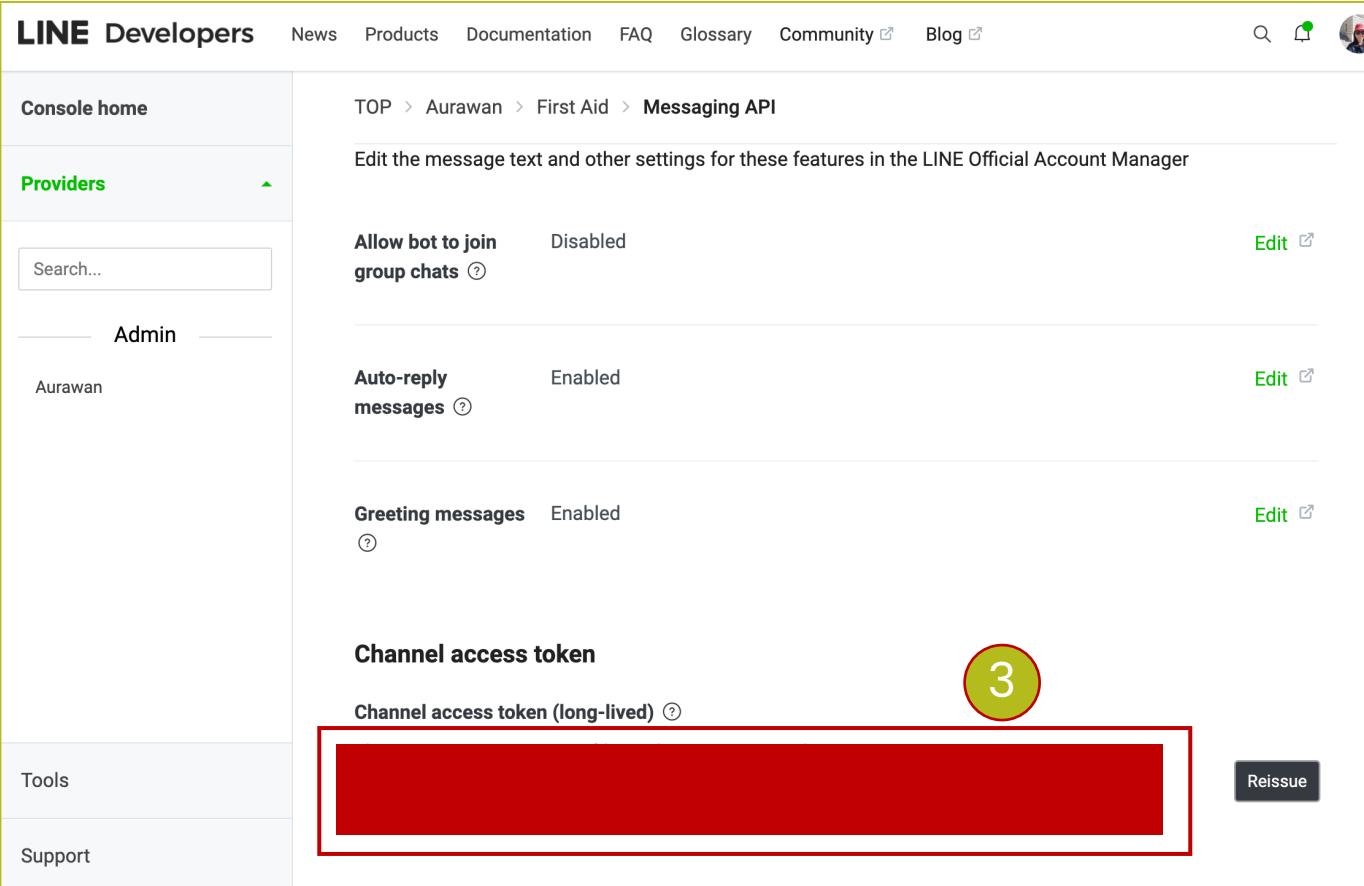
The screenshot shows the LINE Developers console interface. On the left, there's a sidebar with 'Console home', 'Providers' (selected), 'Search...', 'Admin', 'Aurawan', 'Tools', and 'Support'. The main area has a breadcrumb path: TOP > Aurawan > First Aid > Messaging API. It displays a 'First Aid' bot icon with 'Admin' status and 'Messaging API' access. Below are tabs for 'Basic settings' (selected), 'Messaging API' (highlighted with a red box), 'LIFF', 'Security', 'Statistics', and 'Roles'. A section titled 'Messaging API settings' contains 'Bot information' (Bot basic ID: @009stmhr) and a QR code.

10. Scroll down to the Channel access token. Click the Issue button.

This screenshot shows the 'Messaging API' settings page. It includes sections for 'Allow bot to join group chats' (Disabled, Edit), 'Auto-reply messages' (Enabled, Edit), and 'Greeting messages' (Enabled, Edit). Below is a 'Channel access token' section with a sub-section for 'Channel access token (long-lived)' (Issue button highlighted with a red box).

Data for used LINE with Dialogflow

11. Channel access token will be displayed for use in further work.



The screenshot shows the LINE Developers console interface for the Messaging API settings. On the left, there's a sidebar with 'Console home', 'Providers' (selected), a search bar, and sections for 'Admin' (Aurawan) and 'Tools/Support'. The main content area shows three configuration items: 'Allow bot to join group chats' (Disabled, Edit), 'Auto-reply messages' (Enabled, Edit), and 'Greeting messages' (Enabled, Edit). Below these is a 'Channel access token' section. A red box highlights the 'Channel access token (long-lived)' field, which contains a large redacted string. A green circle with the number '3' is overlaid on the top right corner of this red box. To the right of the red box is a 'Reissue' button.

Add LINE friends by using QR code

TOP > Aurawan > First Aid > Messaging API

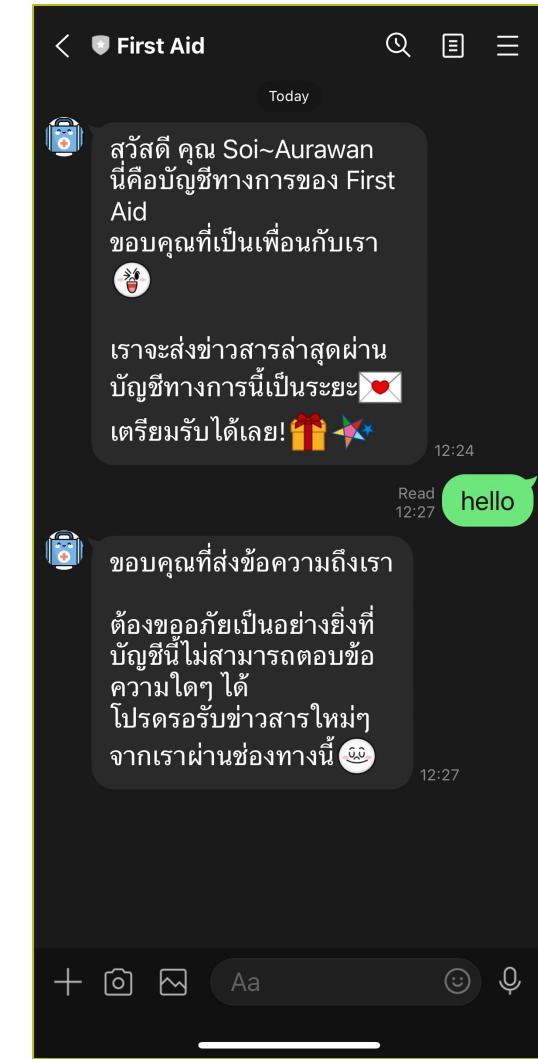
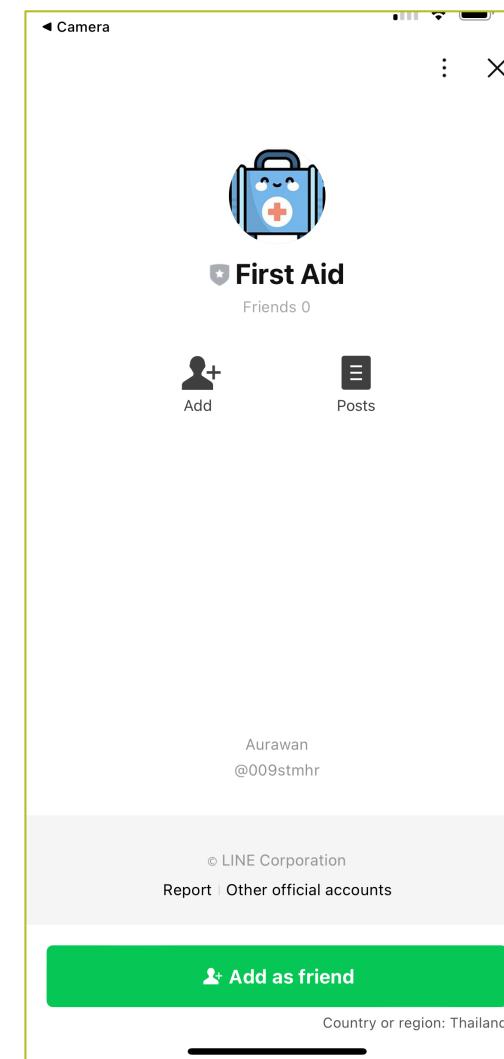
Basic settings **Messaging API** LIFF Security Statistics Roles

Messaging API settings

Bot information

Bot basic ID @009stmhr 

QR code



Disable Auto-reply messages

1. Click Edit in Auto-reply messages section

2. Enabled Webhooks and disabled Auto-response messages

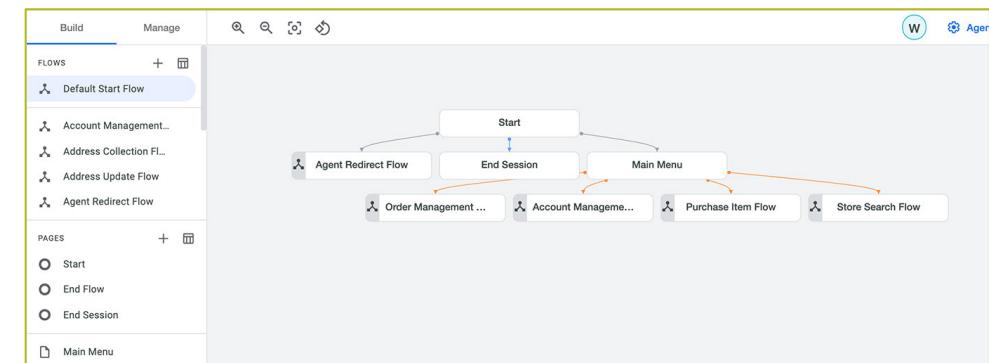
The screenshot shows the LINE Developers console interface. On the left, there's a sidebar with 'Console home', 'Providers' (selected), 'Search...', 'Admin', 'Aurawan', 'Tools', and 'Support'. The main area has a breadcrumb path: TOP > Aurawan > First Aid > Messaging API. A 'Edit' button is visible above the 'LINE Official Account features' section. This section contains settings for 'Allow bot to join group chats' (Disabled) and 'Auto-reply messages' (Enabled). The 'Edit' button for 'Auto-reply messages' is highlighted with a red box. Below this are sections for 'Greeting messages' (Enabled) and 'Channel access token'.

The screenshot shows the 'Response settings' page. It includes a 'Toggle responses' section with four options: 'Chat' (disabled), 'Greeting message' (enabled), 'Webhooks' (enabled), and 'Auto-response messages' (disabled). To the right, a large red text box contains the definition of 'Webhooks': 'Webhooks are automated messages sent from apps when something happens.' Below each toggle is a link to its respective settings page: 'Greeting message settings', 'Messaging API settings', and 'Auto-response message settings'.

Dialogflow

Dialogflow

- Dialogflow is the natural language understanding platform of Google
- Dialogflow allows you to understand language with your chatbot
- Dialogflow has two types of editions:
 - **Dialogflow Essentials (ES) *****
 - The standard edition of Dialogflow
 - Easy to use
 - **Dialogflow Customer Experience (cx)**
 - Has a visual flow builder



Lots of integrations...

 Web Demo	 Dialogflow Messenger BETA	 Messenger from Facebook	 Workplace from Facebook BETA	 Kik 	 Skype 	 Spark 	 Twilio IP Messaging 
 Google Chat	 Slack	 Telegram	 LINE  draft	 Twilio (Text Messaging) 	 Twitter 	 Viber 	

Support many languages include Thai

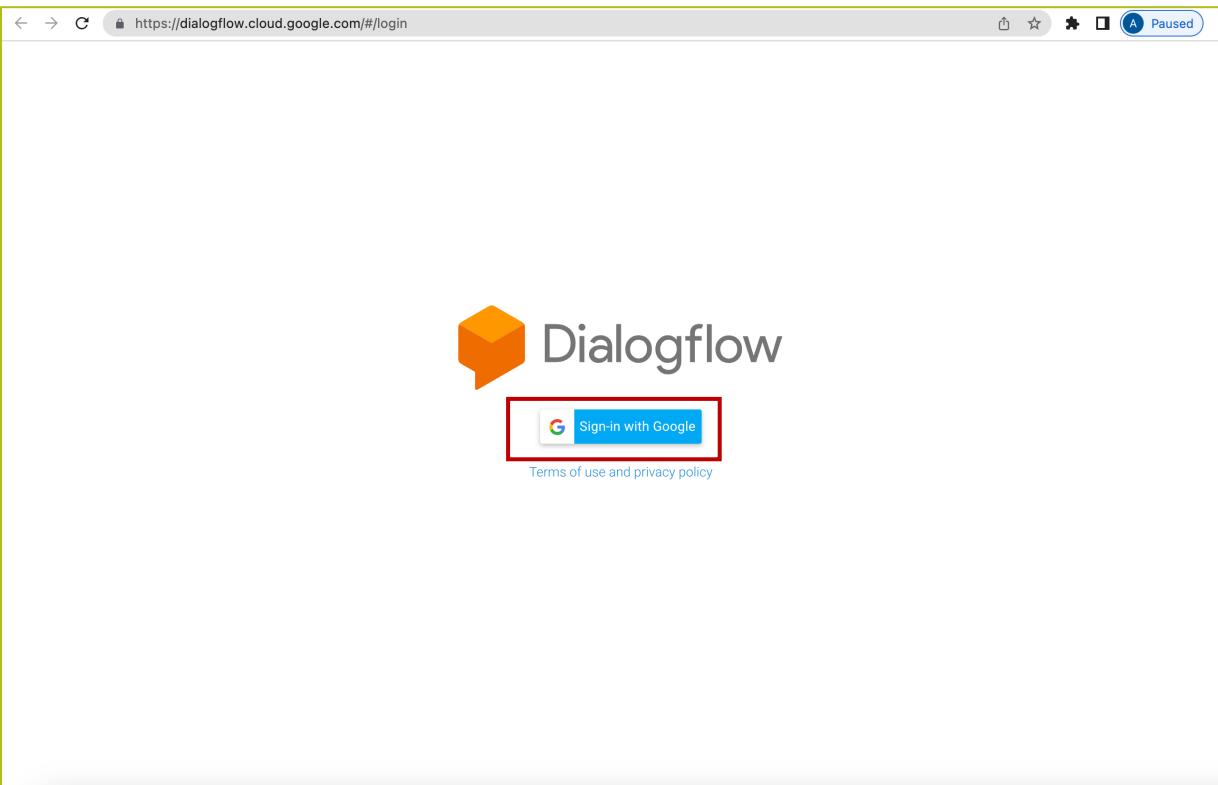
Name	Tag *	Text	STT	TTS	Phone	Knowledge	Sentiment	SmTalk
Afrikaans (Preview)	af	✓	✓	✓				
Albanian (Preview)	sq	✓						
Amharic (Preview)	am	✓						
Armenian (Preview)	hy	✓						
Azerbaijani (Preview)	az	✓						
Basque (Preview)	eu	✓						
Belarusian (Preview)	be	✓						
Bengali	bn	✓						
Bengali - Bangladesh	bn-bd	✓						
Bengali - India	bn-in	✓						
Bosnian (Preview)	bs	✓						
Bulgarian (Preview)	bg	✓	✓	✓				
Catalan (Preview)	ca	✓	✓	✓				
Cebuano (Preview)	ceb	✓						
Chichewa (Preview)	ny	✓						
Chinese - Cantonese	zh-hk	✓	✓	✓				
Chinese - Simplified	zh-cn	✓	✓	✓	✓		✓	



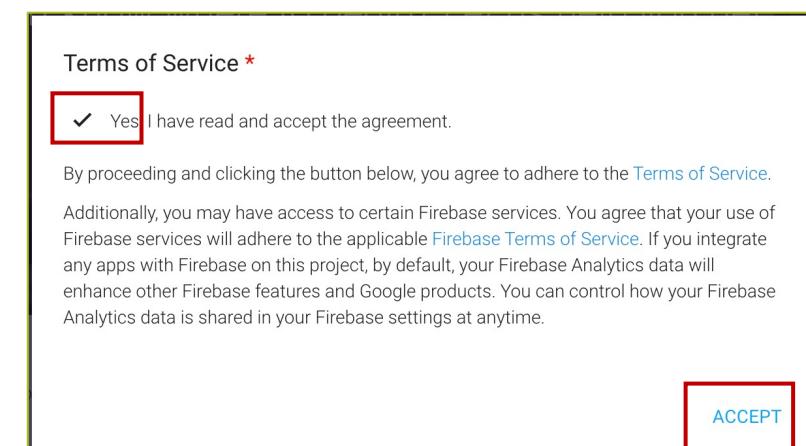
Dialogflow

1. goto

<https://dialogflow.cloud.google.com/>



2. Sign in with Google account
and accept the agreement



A screenshot of a Terms of Service acceptance dialog box. It contains the following text:

Terms of Service *

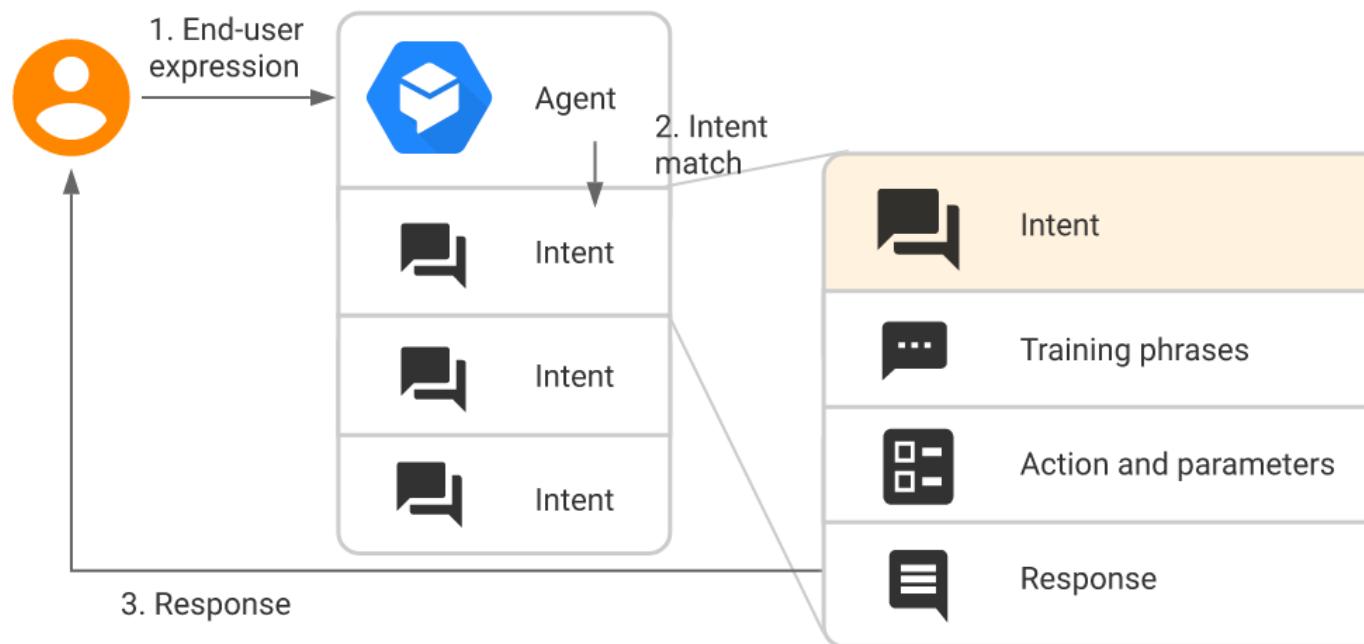
Yes, I have read and accept the agreement.

By proceeding and clicking the button below, you agree to adhere to the [Terms of Service](#). Additionally, you may have access to certain Firebase services. You agree that your use of Firebase services will adhere to the applicable [Firebase Terms of Service](#). If you integrate any apps with Firebase on this project, by default, your Firebase Analytics data will enhance other Firebase features and Google products. You can control how your Firebase Analytics data is shared in your Firebase settings at anytime.

ACCEPT

A red rectangular box highlights the "ACCEPT" button.

Dialogflow

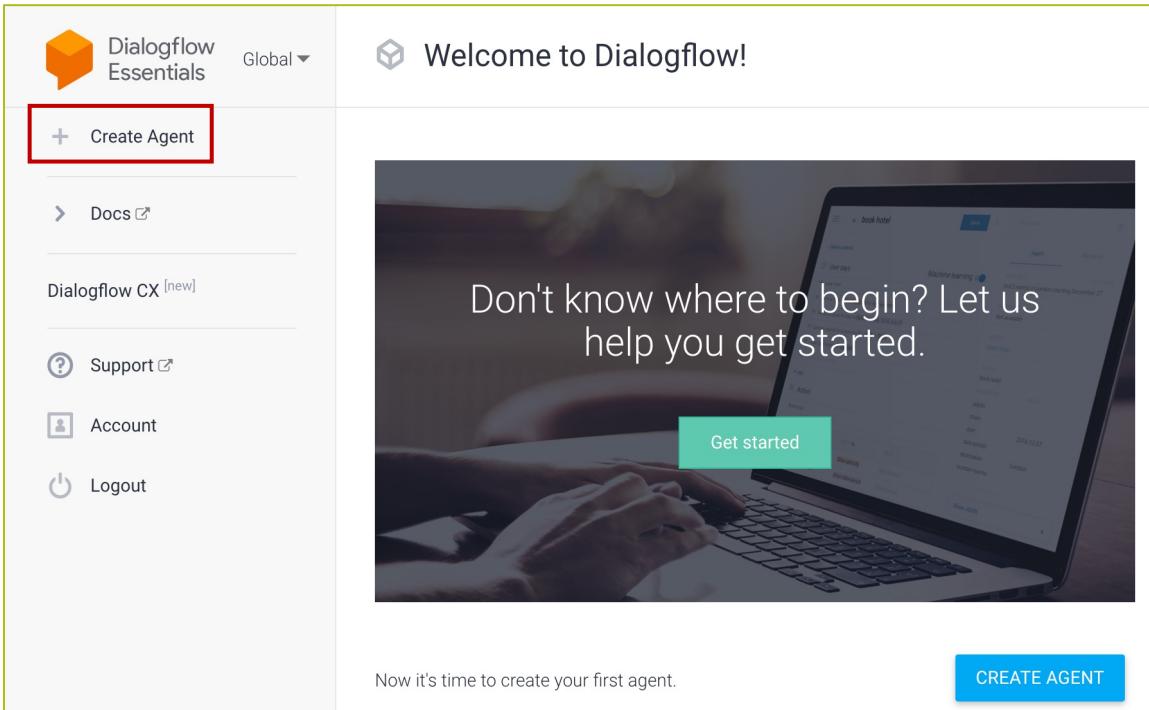


An **agent** is a virtual agent that handles concurrent conversations with the end-users.

An **intent** categorizes an end-user's intention for one conversation turn.

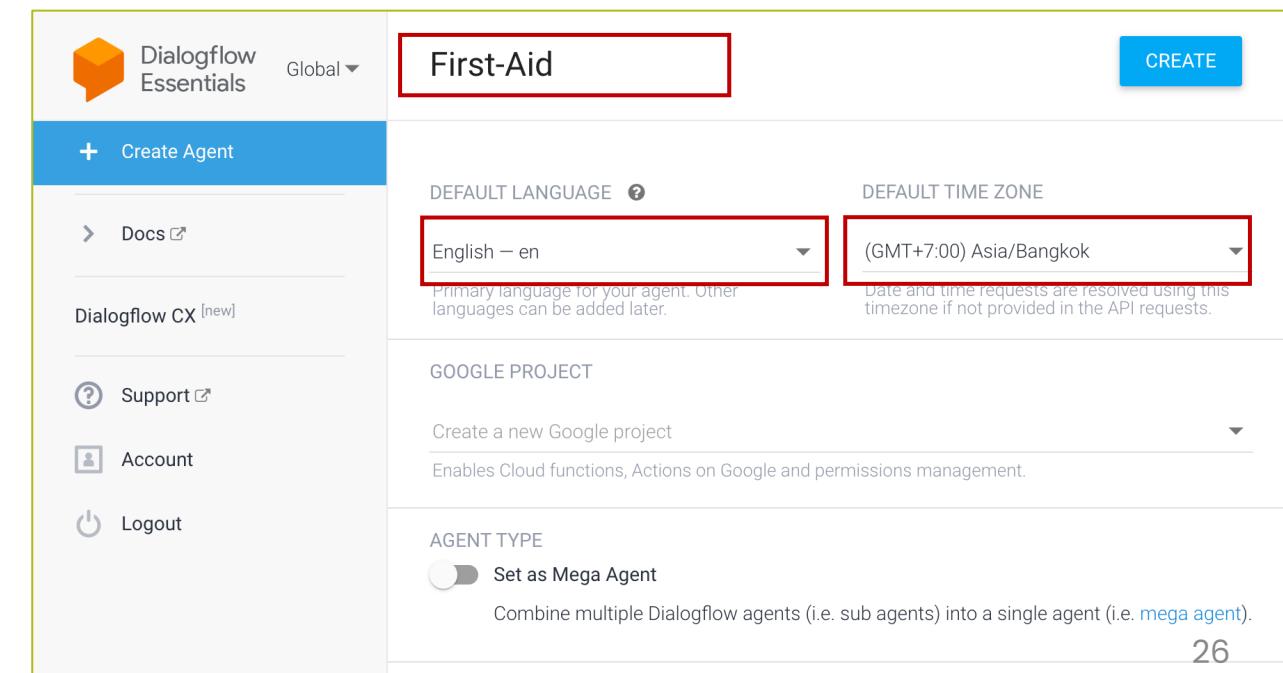
Create Agent

3. Click Create Agent



The screenshot shows the Dialogflow Essentials dashboard. On the left, there's a sidebar with navigation links: 'Docs', 'Dialogflow CX [new]', 'Support', 'Account', and 'Logout'. The main area has a title 'Welcome to Dialogflow!' and a large image of a person typing on a laptop. A green 'Get started' button is visible. At the bottom, a blue 'CREATE AGENT' button is highlighted with a red box. The overall background is white with a light gray header.

4. Enter Agent name,
Select default language,
default time-zone
then click Create button

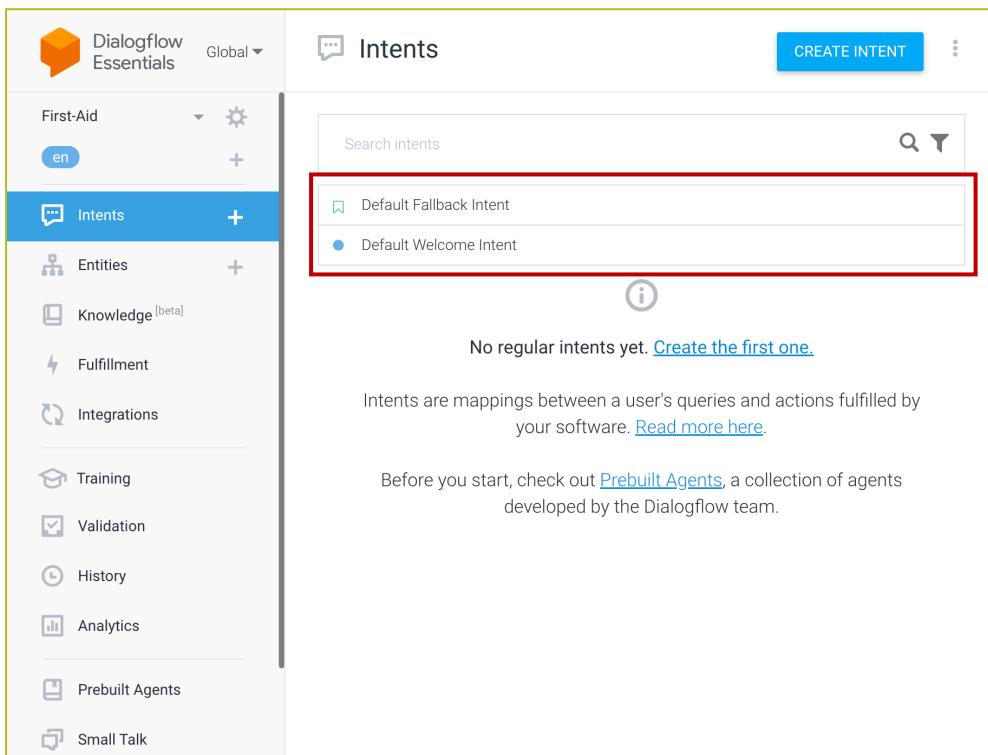


The screenshot shows the 'Create Agent' form. It includes fields for 'Agent name' (containing 'First-Aid'), 'Default Language' (set to 'English – en'), and 'Default Time Zone' ('(GMT+7:00) Asia/Bangkok'). Other sections like 'Google Project' and 'Agent Type' are also visible. The 'CREATE' button is at the top right. The entire form is enclosed in a light gray box with a yellow border.

Default Intent

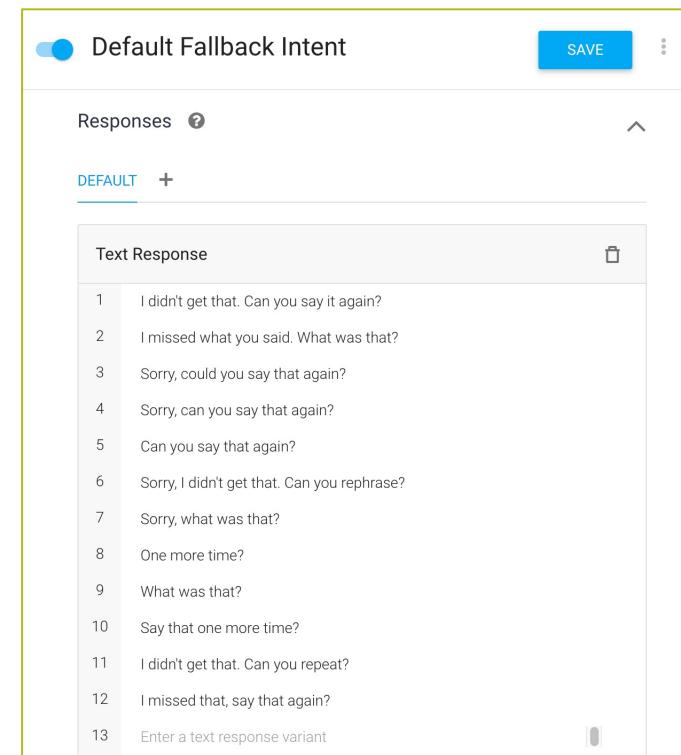
Intent

The **default fallback intent** is matched when your agent doesn't recognize an end-user expression.

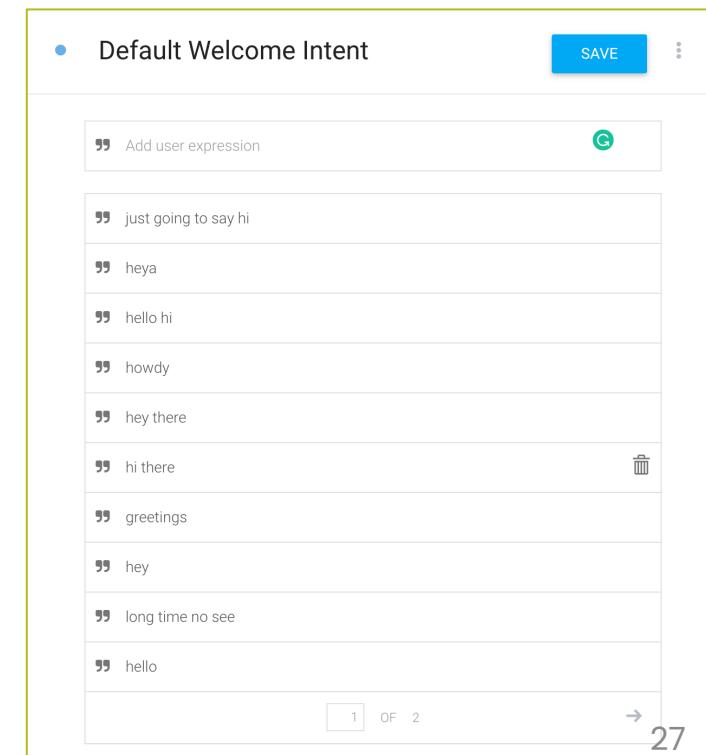


The screenshot shows the Dialogflow interface with the 'Intents' tab selected in the sidebar. A red box highlights the 'Default Fallback Intent' listed under the 'Responses' section. The interface includes a search bar, a 'CREATE INTENT' button, and a sidebar with various project management and fulfillment options.

The **default welcome intent** is matched when an end-user begins a conversation with your agent.

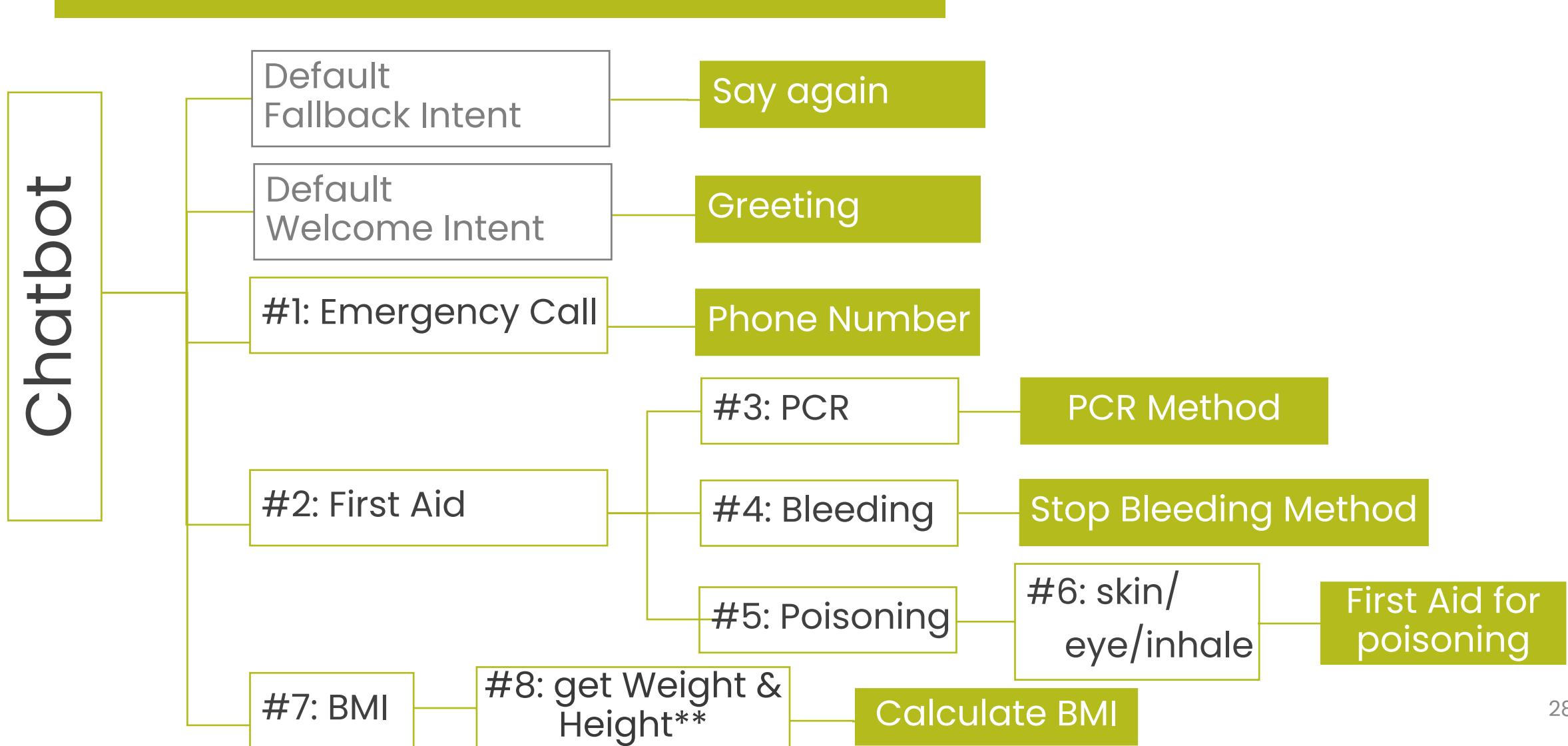


The screenshot shows the 'Default Fallback Intent' configuration page. It features a 'Responses' section with a 'Text Response' card containing a list of 13 generic text responses such as 'I didn't get that. Can you say it again?' and 'Sorry, what was that?'. A 'DEFAULT' tab is selected at the bottom.



The screenshot shows the 'Default Welcome Intent' configuration page. It lists several user expressions mapped to a single response: 'just going to say hi', 'heya', 'hello hi', 'howdy', 'hey there', 'hi there', 'greetings', 'hey', 'long time no see', and 'hello'. A 'SAVE' button is visible at the top right.

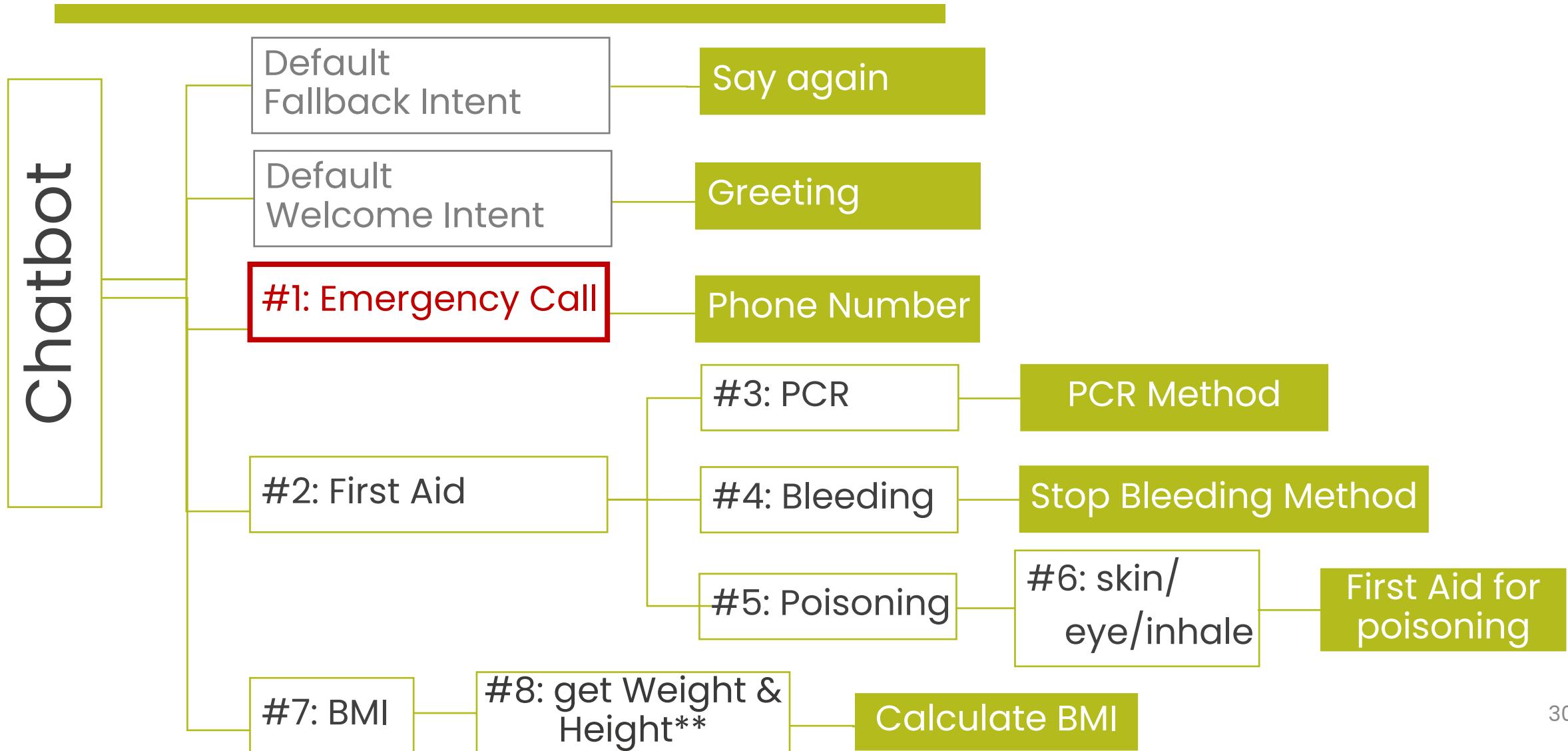
Intent Mapping



Intent#1

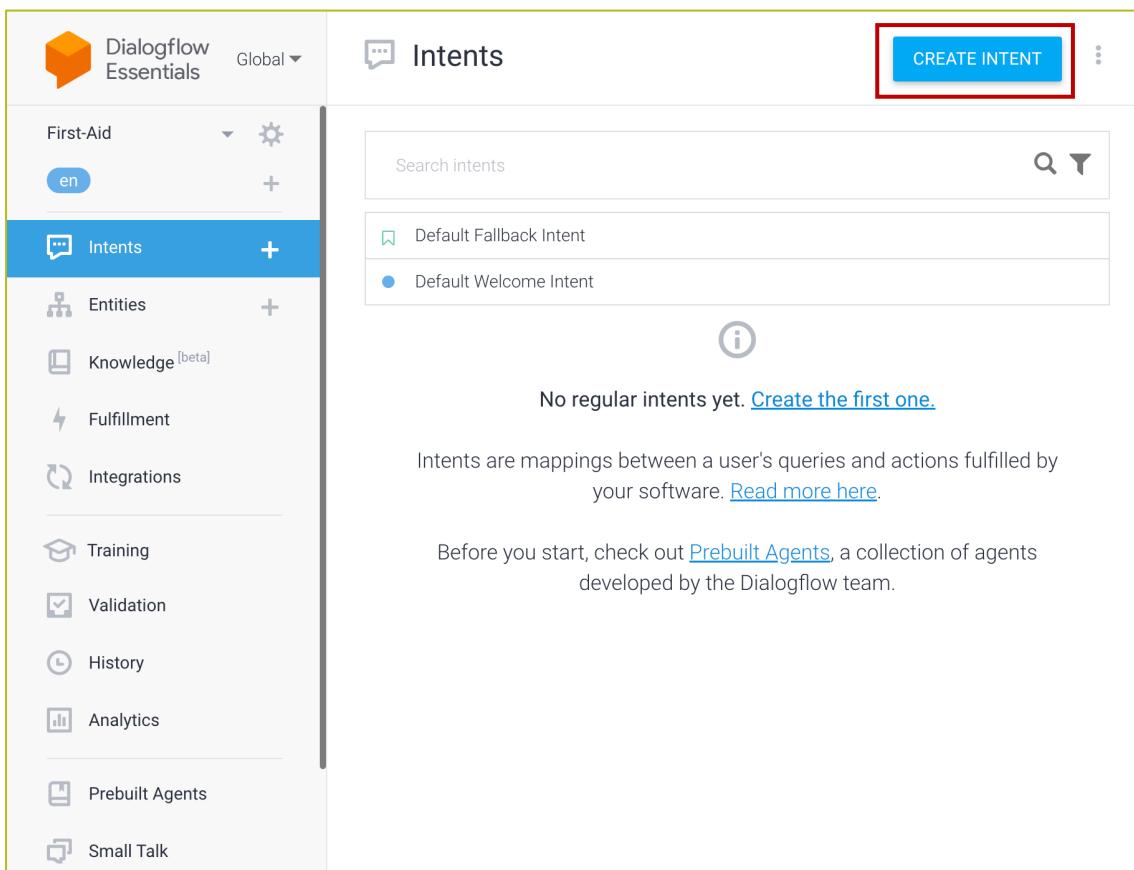
Emergency Call

Intent Mapping



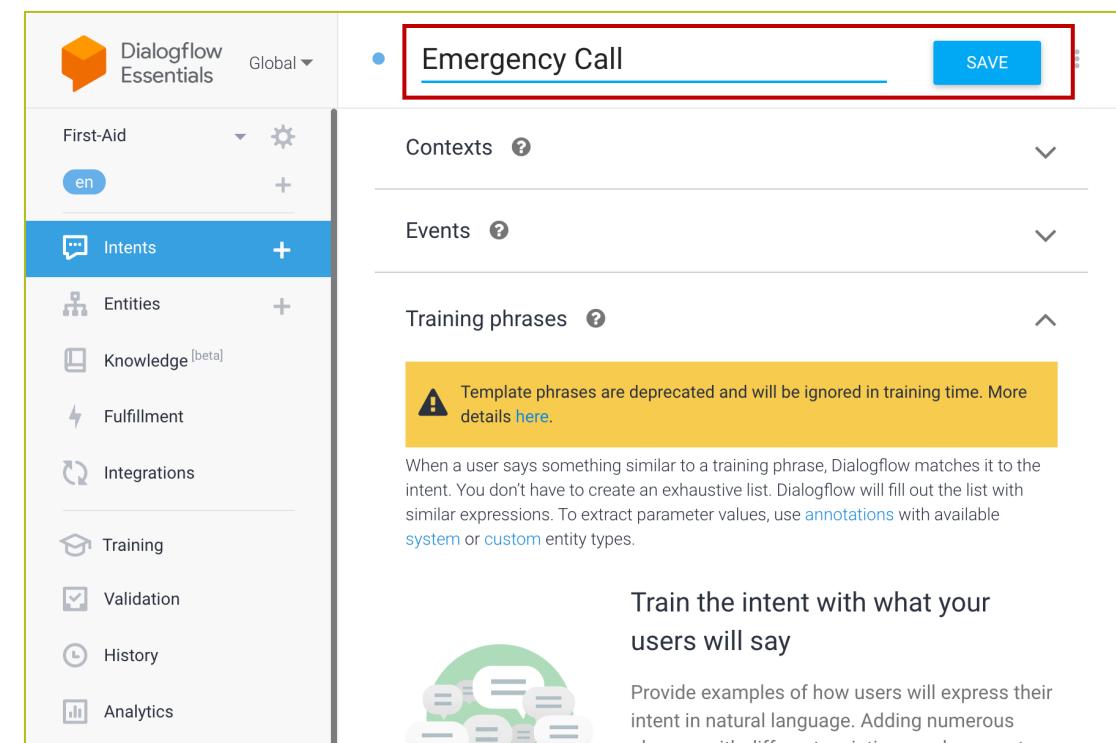
Create Intent#1

1. Click Create Intent



The screenshot shows the Dialogflow Essentials dashboard. On the left, a sidebar lists various project components: First-Aid (selected), Entities, Knowledge [beta], Fulfillment, Integrations, Training, Validation, History, Analytics, Prebuilt Agents, and Small Talk. The 'Intents' tab is also selected in the sidebar. The main content area is titled 'Intents' and contains a 'CREATE INTENT' button highlighted with a red box. Below it is a search bar labeled 'Search intents' and two listed intents: 'Default Fallback Intent' and 'Default Welcome Intent'. A message at the bottom says 'No regular intents yet. [Create the first one.](#)'

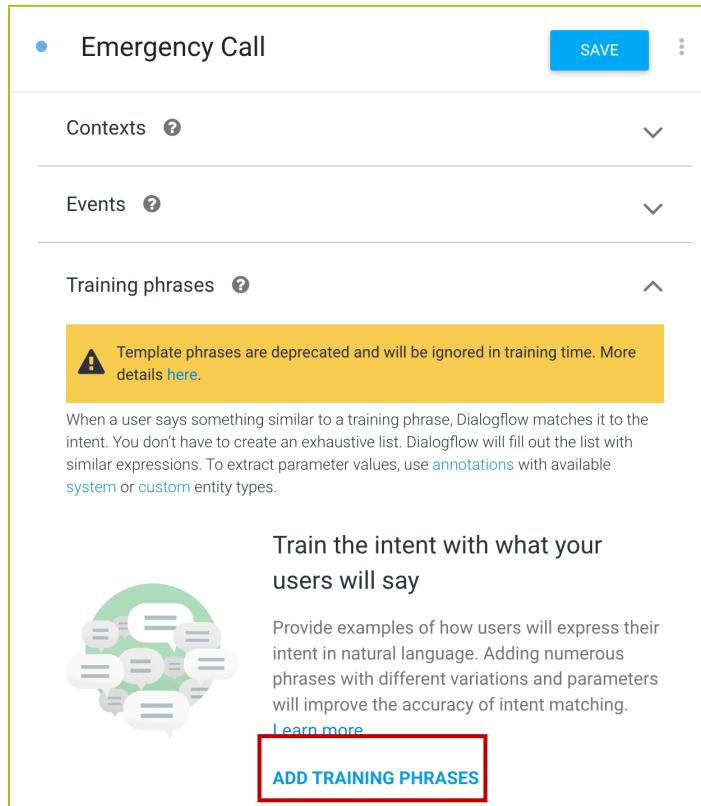
2. Type the new intent name and press the save button.



The screenshot shows the 'Intents' creation screen. The intent name 'Emergency Call' is typed into the input field, which is highlighted with a red box. To the right is a 'SAVE' button. The interface includes sections for 'Contexts', 'Events', and 'Training phrases'. A yellow warning box states: 'Template phrases are deprecated and will be ignored in training time. More details [here](#)'. Below this, a note explains: 'When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use [annotations](#) with available system or custom entity types.' At the bottom, there's a section titled 'Train the intent with what your users will say' with a green icon, and a note: 'Provide examples of how users will express their intent in natural language. Adding numerous phrases with different variations and parameters'.

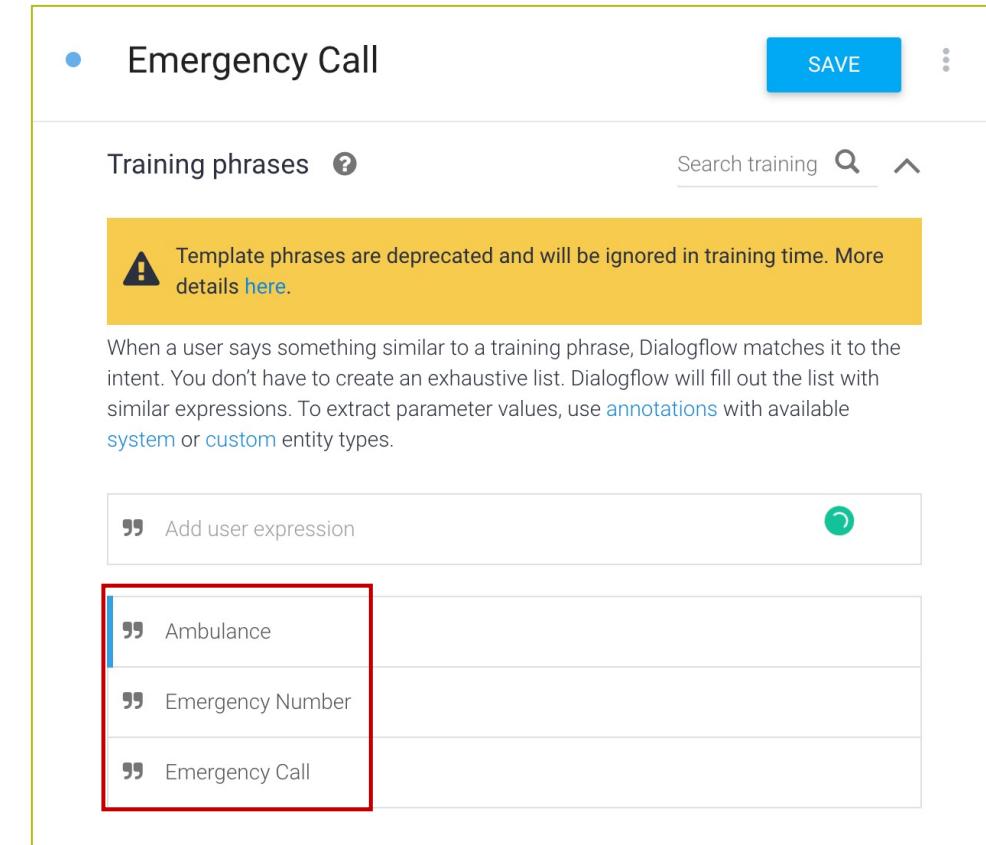
Create Intent#1

3. Click “Add Training Phrases” Text



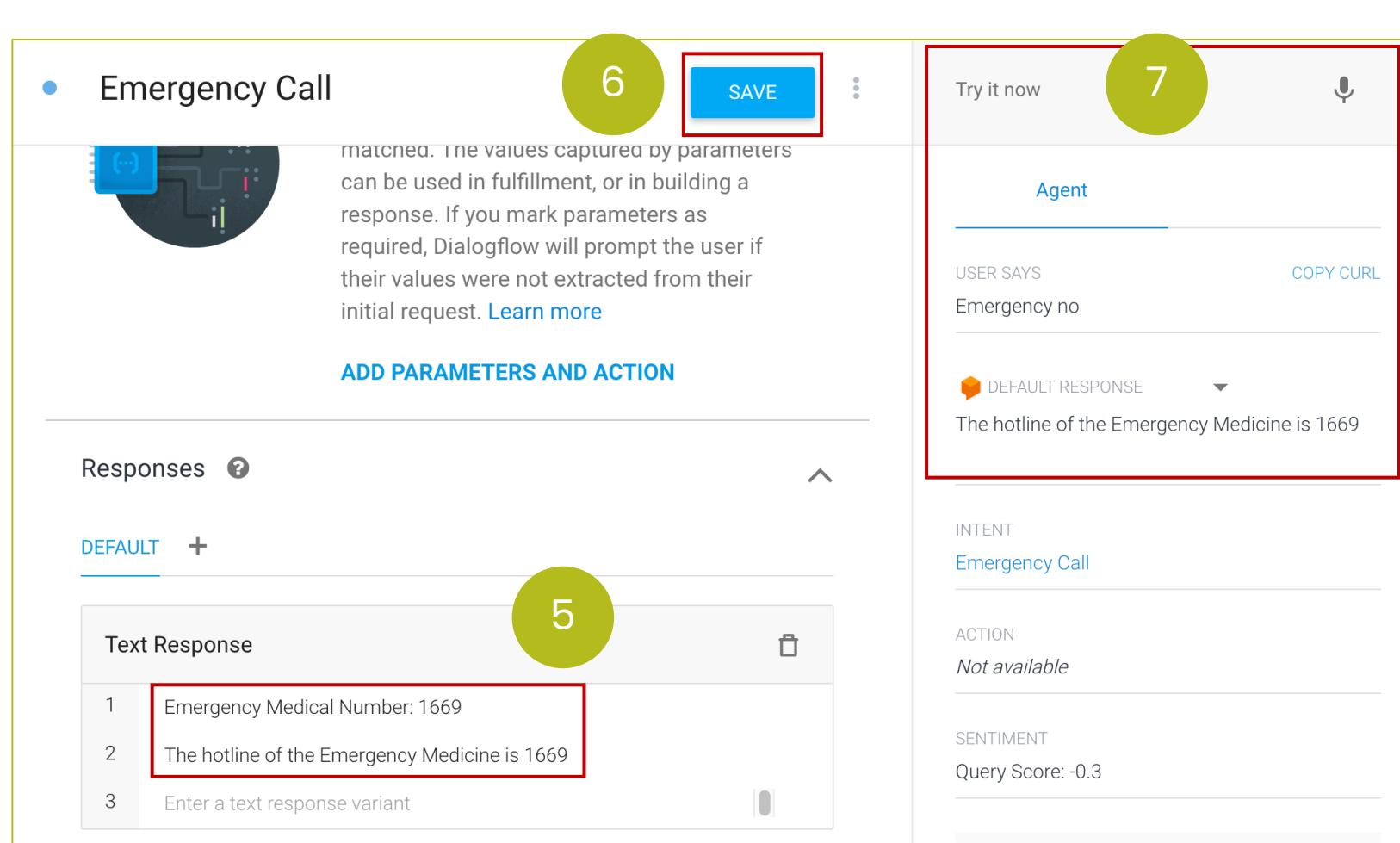
The screenshot shows the Dialogflow Intent creation interface. At the top, there's a list of intents: "Emergency Call". Below it are sections for "Contexts" and "Events", both with dropdown menus. The "Training phrases" section is expanded, showing a yellow warning box: "Template phrases are deprecated and will be ignored in training time. More details [here](#)". Below this, a text block explains that Dialogflow matches user input to training phrases even if they're similar, and provides instructions for extracting parameter values using annotations. A green icon of a speech bubble with multiple lines of text is on the left. A red-bordered button at the bottom right says "ADD TRAINING PHRASES".

4. Type the training phrases of the intent



The screenshot shows the "Training phrases" section for the "Emergency Call" intent. It includes a search bar and a note about template phrases being deprecated. Below is a list of three training phrases: "Ambulance", "Emergency Number", and "Emergency Call", each preceded by a double引号 icon. The last two items are highlighted with a red border. A blue "SAVE" button is at the top right.

Create Intent#1



• Emergency Call

matchnea. The values captured by parameters can be used in fulfillment, or in building a response. If you mark parameters as required, Dialogflow will prompt the user if their values were not extracted from their initial request. [Learn more](#)

ADD PARAMETERS AND ACTION

Responses ?

DEFAULT +

Text Response

5

1 Emergency Medical Number: 1669
2 The hotline of the Emergency Medicine is 1669
3 Enter a text response variant

6 SAVE

Try it now

7

Agent

USER SAYS COPY CURL

Emergency no

DEFAULT RESPONSE

The hotline of the Emergency Medicine is 1669

INTENT

Emergency Call

ACTION

Not available

SENTIMENT

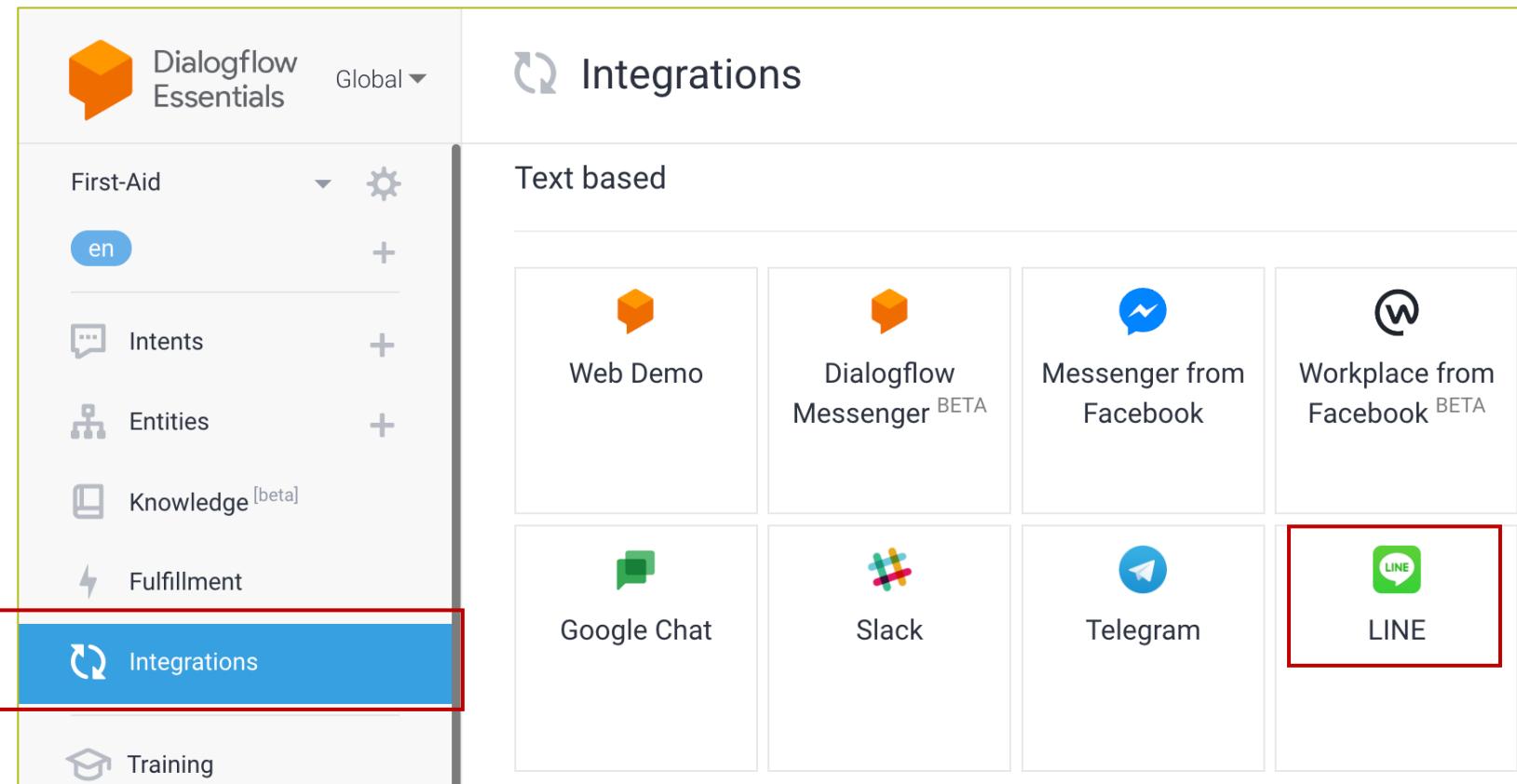
Query Score: -0.3

5. Add Text Response
6. Press Save button
7. Test within Dialogflow

Connect Dialogflow to LINE

Connect Dialogflow to LINE

1. Click Integrations and click Line icon



Connect Dialogflow to LINE

2. copy Channel ID, Channel Secret and Channel Access Token from LINE to Dialogflow and click Start button

 Line
Free Messaging, Whenever, Wherever.

- In the 'Messaging API' section, click 'LINE Developers' to go to the Channel Console.
- Copy Channel ID and Channel Secret and paste into the respective fields below.
- Click 'ISSUE' for the 'Channel access token' item and paste its value to the respective field below.
- Click 'EDIT' and set the Webhook URL for your Channel by copying and pasting its value from the field below.
- Then click 'SAVE' and 'VERIFY'.
- Click the 'START' button below.

More in documentation.

Channel ID	[REDACTED]
Channel Secret	[REDACTED]
Channel Access Token	[REDACTED]
Webhook URL	https://dialogflow.cloud.google.com/v1/integrations/line/webhook/020b71bd-9fd6-4a2b-a95

Choose an environment to use with this integration.

Environment: Draft

CLOSE **START**

LINE Developers News Products Documentation FAQ Glossary Community Blog Search

Console home Providers Search... Admin Aurawan Tools

TOP > Aurawan > First Aid > Basic settings

Basic settings Messaging API LIFF Security Statistics Roles

Basic settings

Basic information

You can change your app name and icon in [LINE Official Account Manager](#).

Channel ID [REDACTED] Channel icon 

Permissions PROFILE Channel secret [REDACTED] Issue

Assertion Signing Key Register a public key

Messaging API

Allow bot to join group chats Disabled Edit

Auto-reply messages Disabled Edit

Greeting messages Enabled Edit

Channel access token

Channel access token (long-lived) [REDACTED] Reissue

Connect Dialogflow to LINE

3. copy Webhook URL from Dialogflow to LINE and enable to use webhook

Line
Free Messaging, Whenever, Wherever.

Build an intelligent conversational LINE bot.

When your Dialogflow agent is ready, follow these instructions to connect it to a LINE Channel:

- If you don't have a LINE@ account, [create a LINE@ account with the Messaging API enabled](#).
- In the [LINE@ Manager](#), go to Settings > Bot Settings from the left side menu.
- On the Bot Settings page, in the 'Request Settings' section, set 'Allow' for 'Use webhooks'.
- Go to your LINE@ account page in the [LINE Business Center](#).
- In the 'Messaging API' section, click 'LINE Developers' to go to the Channel Console.
- Copy Channel ID and Channel Secret and paste into the respective fields below.
- Click 'ISSUE' for the 'Channel access token' item and paste its value to the respective field below.
- Click 'EDIT' and set the Webhook URL for your Channel by copying and pasting its value from the field below. Then click 'SAVE' and 'VERIFY'.
- Click the 'START' button below.

More in documentation.

Channel ID: [REDACTED]

Channel Secret: [REDACTED]

Channel Access Token: [REDACTED]

Webhook URL: https://dialogflow.cloud.google.com/v1/integrations/line/webhook/020b71bd-9fd6-4a2b-a959-dfec8faac6c1

LINE Developers News Products Documentation FAQ Glossary Community Blog Search

Console home Providers Search... Admin Aurawan

TOP > Aurawan > First Aid > Messaging API

Webhook settings

Webhook URL ? Edit 3.2

LINE Official Account features

Search... Admin Aurawan

Webhook settings

Webhook URL ? https://dialogflow.cloud.google.com/v1/integrations/line/webhook/020b71bd-9fd6-4a2b-a959-dfec8faac6c1 3.3

✓ Don't leave this empty
✓ Enter a valid HTTPS URL
✓ Enter no more than 500 characters

Update Cancel

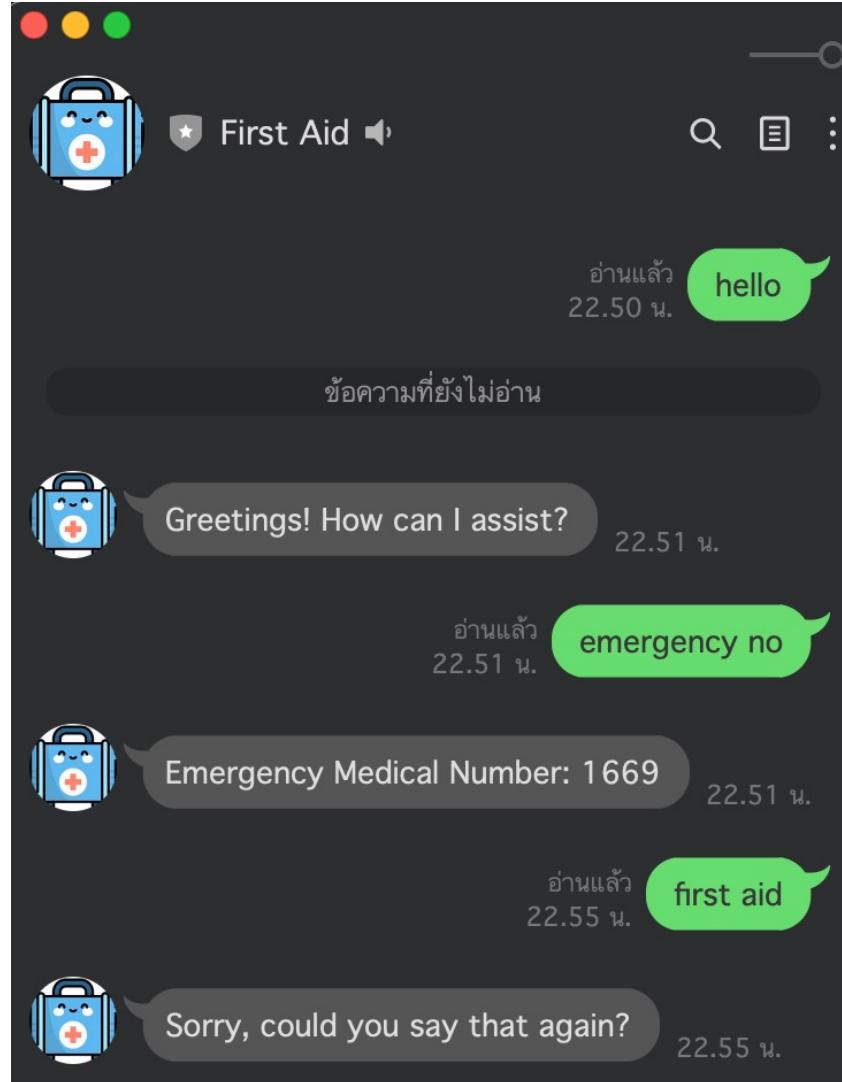
Webhook settings

Webhook URL ? https://dialogflow.cloud.google.com/v1/integrations/line/webhook/020b71bd-9fd6-4a2b-a959-dfec8faac6c1 Verify Edit 3.4

Use webhook ? 3.5

Connect Dialogflow to LINE

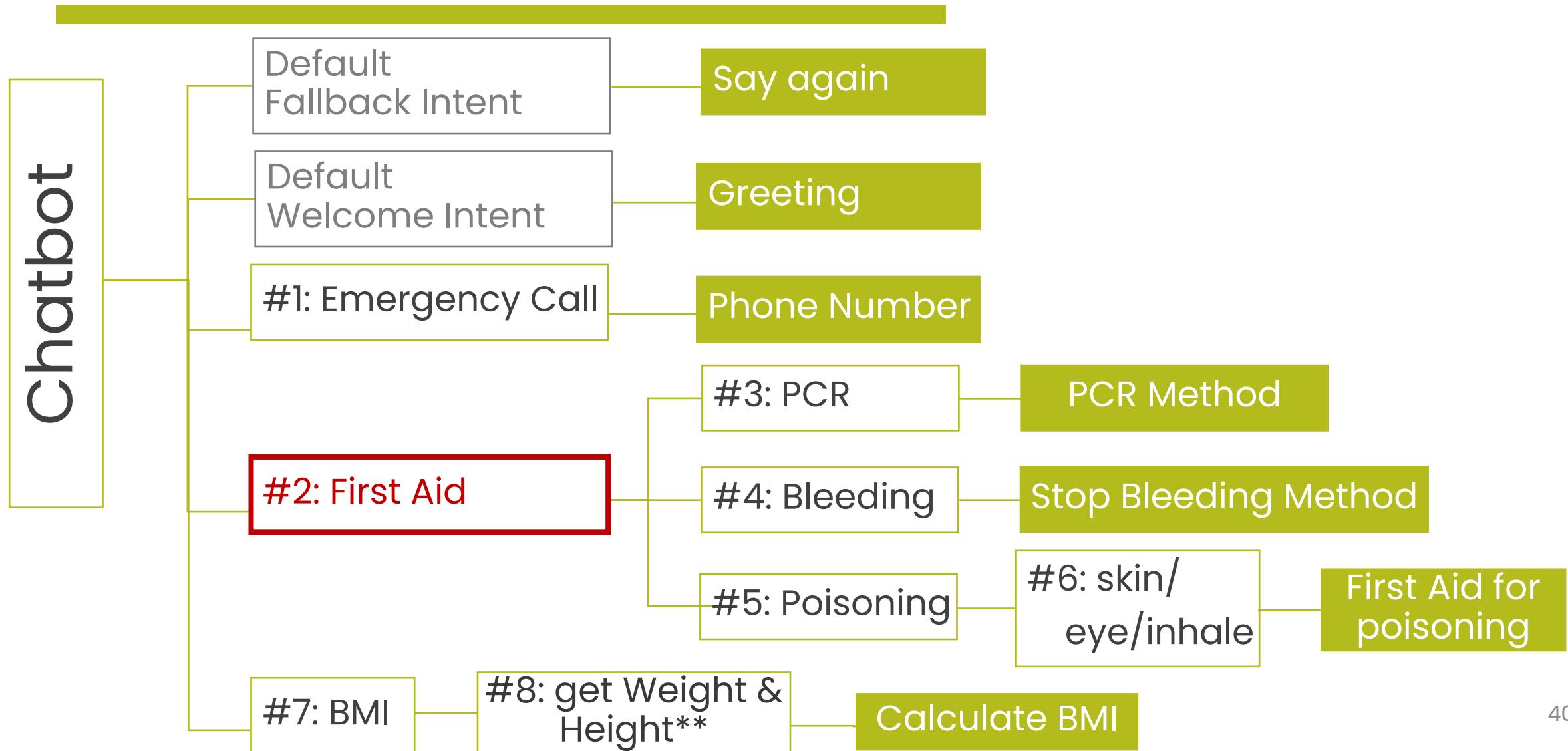
4. Test with LINE Application



Intent#2 & #3

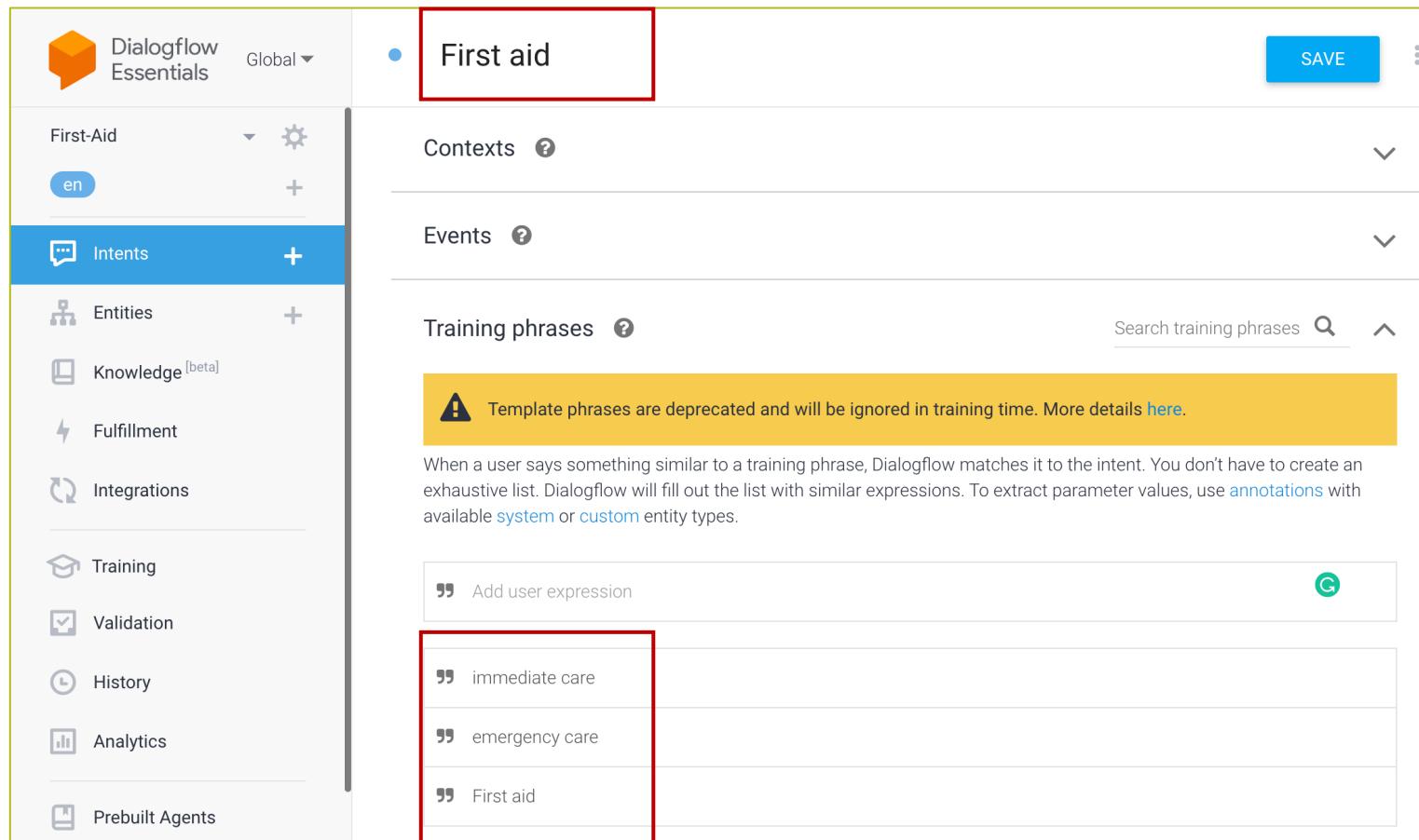
First Aid & PCR

Intent Mapping



Intent#2

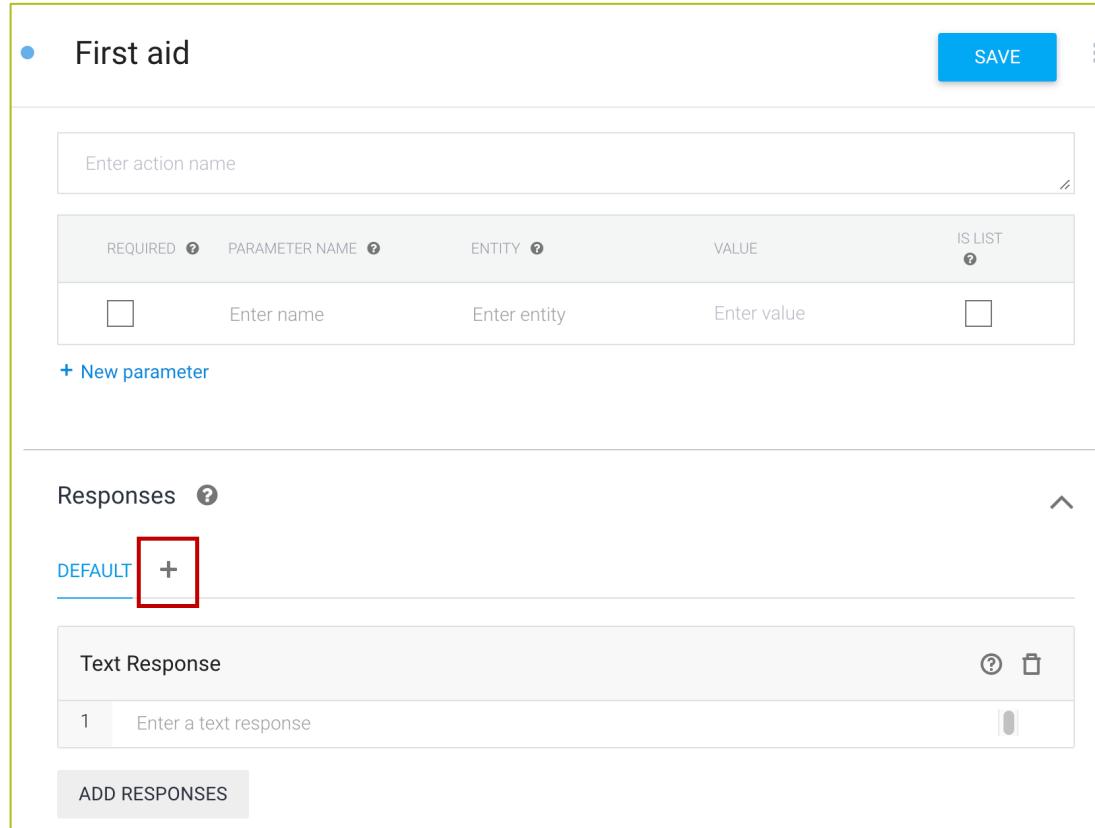
1. Type First aid intent name and Training phrases



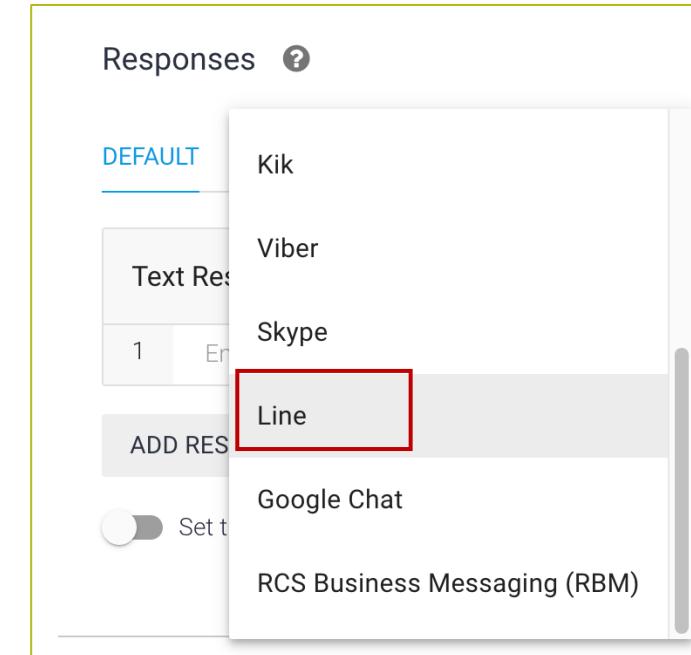
The screenshot shows the Dialogflow Essentials interface. On the left, a sidebar lists various sections: First-Aid (selected), Entities, Knowledge [beta], Fulfillment, Integrations, Training, Validation, History, Analytics, and Prebuilt Agents. The main area is titled 'First aid'. It includes sections for 'Contexts', 'Events', and 'Training phrases'. A note at the top of the training phrases section states: 'Template phrases are deprecated and will be ignored in training time. More details [here](#)'. Below this, a paragraph explains that Dialogflow matches user expressions to training phrases even if they're not exact. The 'Training phrases' list contains three entries: 'immediate care', 'emergency care', and 'First aid'. The entire 'First aid' section is highlighted with a red box.

Intent#2

2. In Response Section click + and select LINE from the list



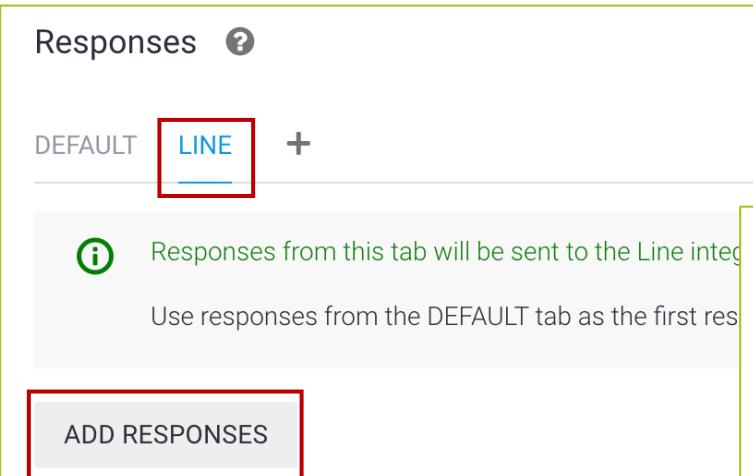
The screenshot shows the 'Responses' section of the Dialogflow Intent editor for the 'First aid' intent. The 'DEFAULT' tab is selected. A red box highlights the '+' button used to add new response types. Below it, a 'Text Response' card is visible, showing a numbered list item '1' and a placeholder 'Enter a text response'. At the bottom of the responses section is a 'ADD RESPONSES' button.



The screenshot shows the 'Responses' list in the Dialogflow Intent editor. The 'Line' option is highlighted with a red box. Other listed platforms include Kik, Viber, Text Response, Skype, Google Chat, and RCS Business Messaging (RBM). A 'Set t...' toggle switch is visible at the bottom left of the list.

Intent#2 & #3

3. Select LINE Tab, Press Add Responses Button and Select Card from the list



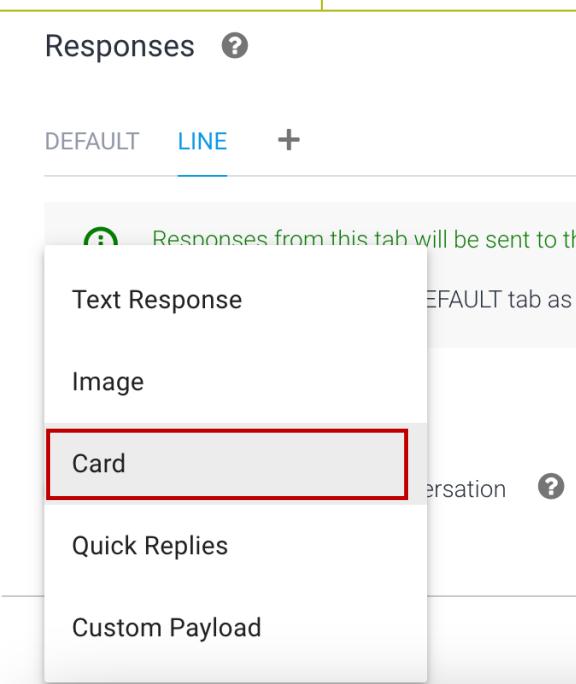
Responses ?

DEFAULT LINE +

i Responses from this tab will be sent to the Line integ...

Use responses from the DEFAULT tab as the first res...

ADD RESPONSES



Responses ?

DEFAULT LINE +

i Responses from this tab will be sent to the Line integ...

Use responses from the DEFAULT tab as the first res...

Text Response

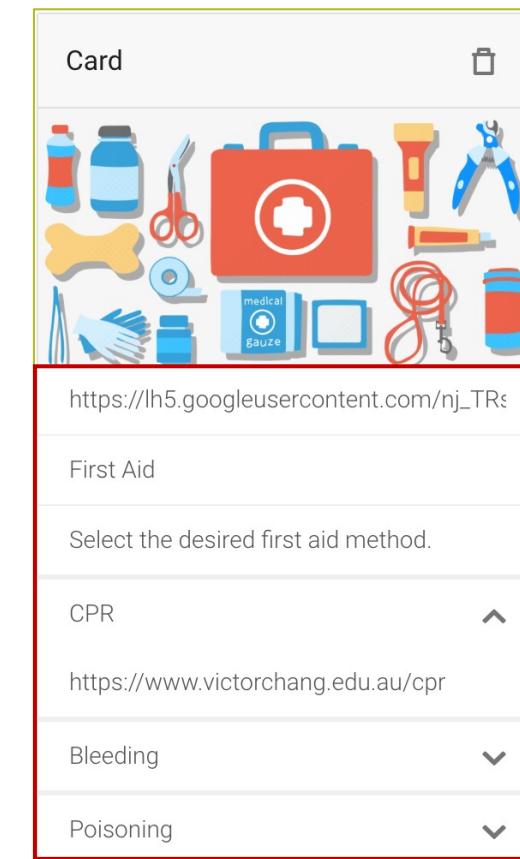
Image

Card

Quick Replies

Custom Payload

4. Select LINE Tab, Press Add Responses Button and Select Card from the list



Card ?



https://lh5.googleusercontent.com/nj_TRsCL1GBOkvEi75y1r82Np-Dt06t-VTDddB7eD8u20nlnxz_tI438n5jYjq61hI0qDmwD2ABpAWWLKQfRSiE

First Aid

Select the desired first aid method.

CPR

<https://www.victorchang.edu.au/cpr>

Bleeding

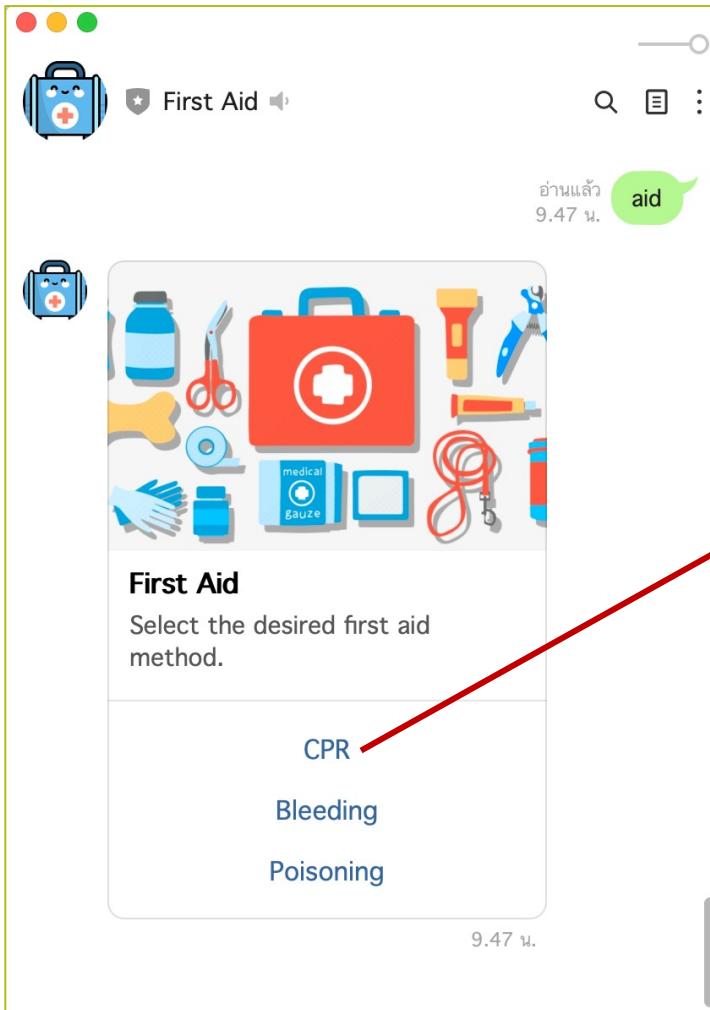
Poisoning

example link for image

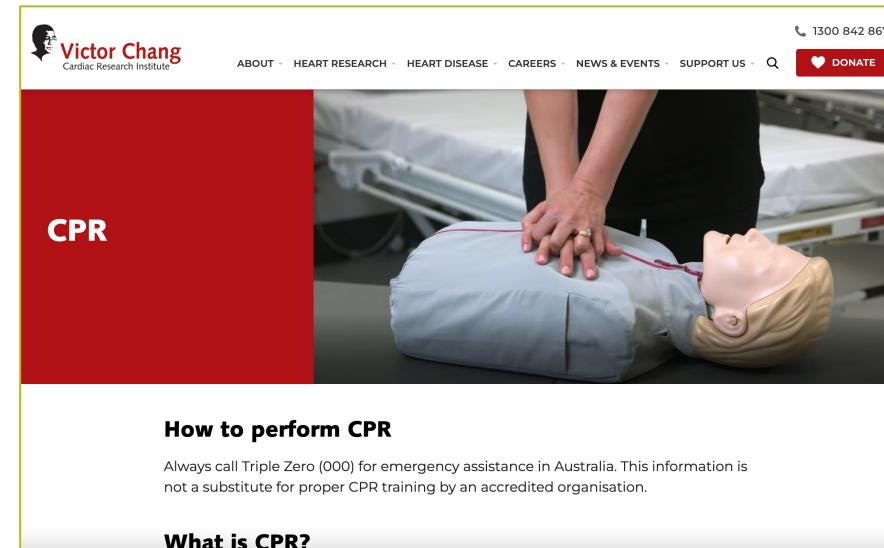
https://lh5.googleusercontent.com/nj_TRsCL1GBOkvEi75y1r82Np-Dt06t-VTDddB7eD8u20nlnxz_tI438n5jYjq61hI0qDmwD2ABpAWWLKQfRSiE

Intent#2 & #3

5. Test with LINE Application



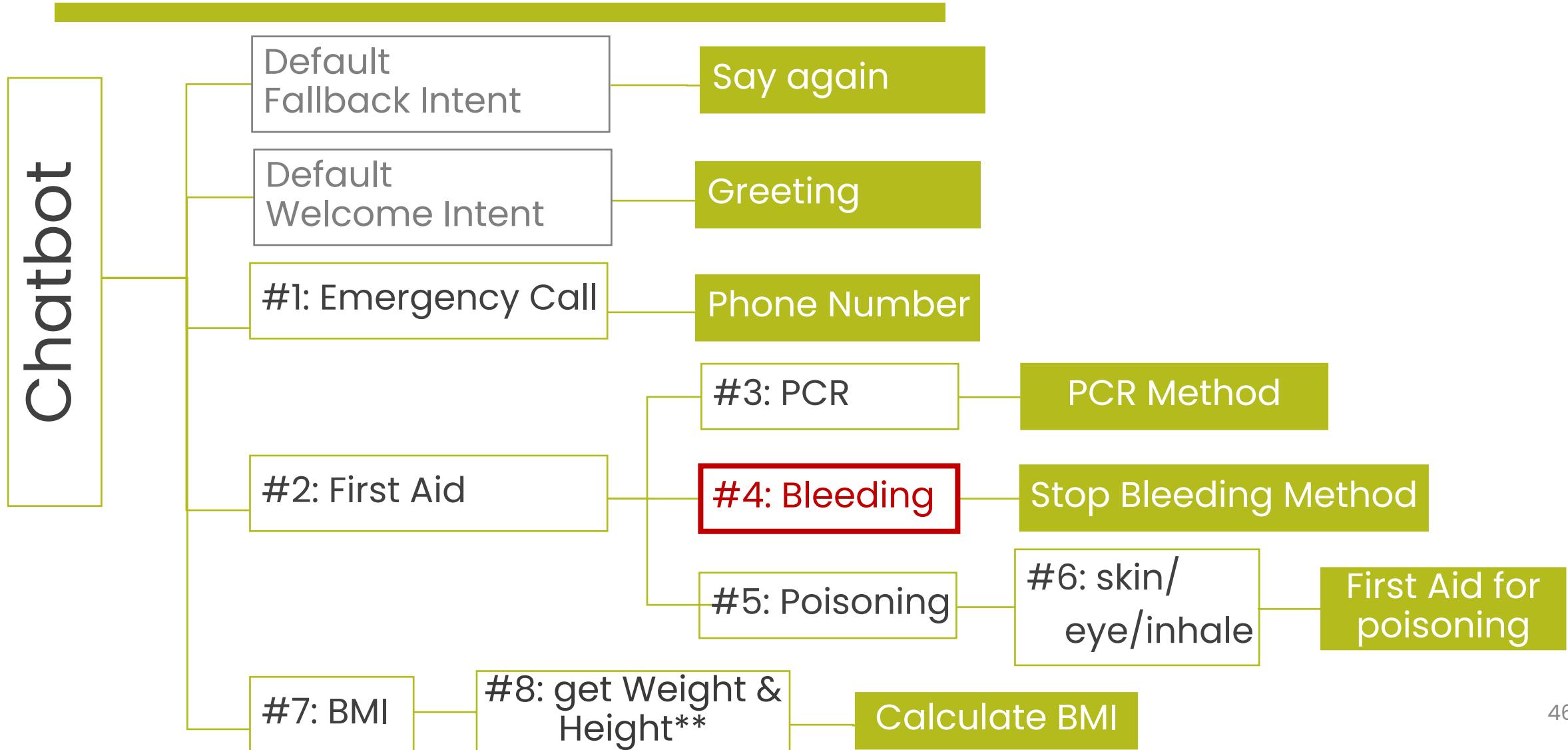
Open Web:
<https://www.victorchang.edu.au/cpr>



Intent#4

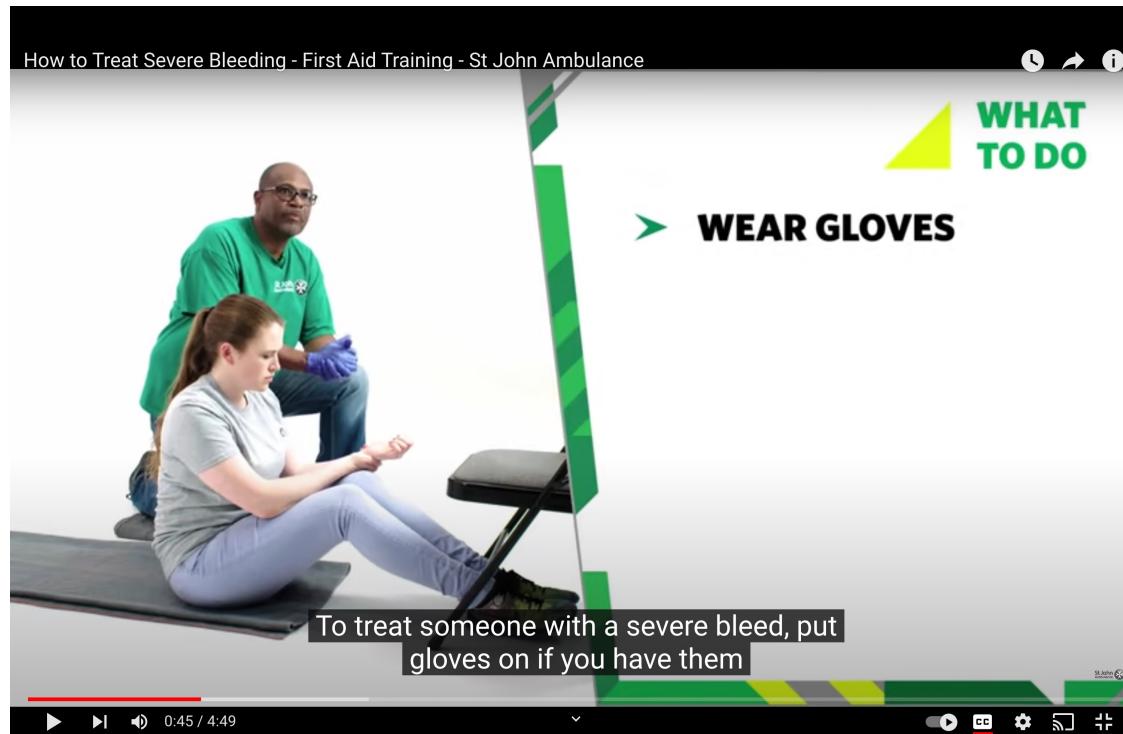
Bleeding

Intent Mapping



Intent#4

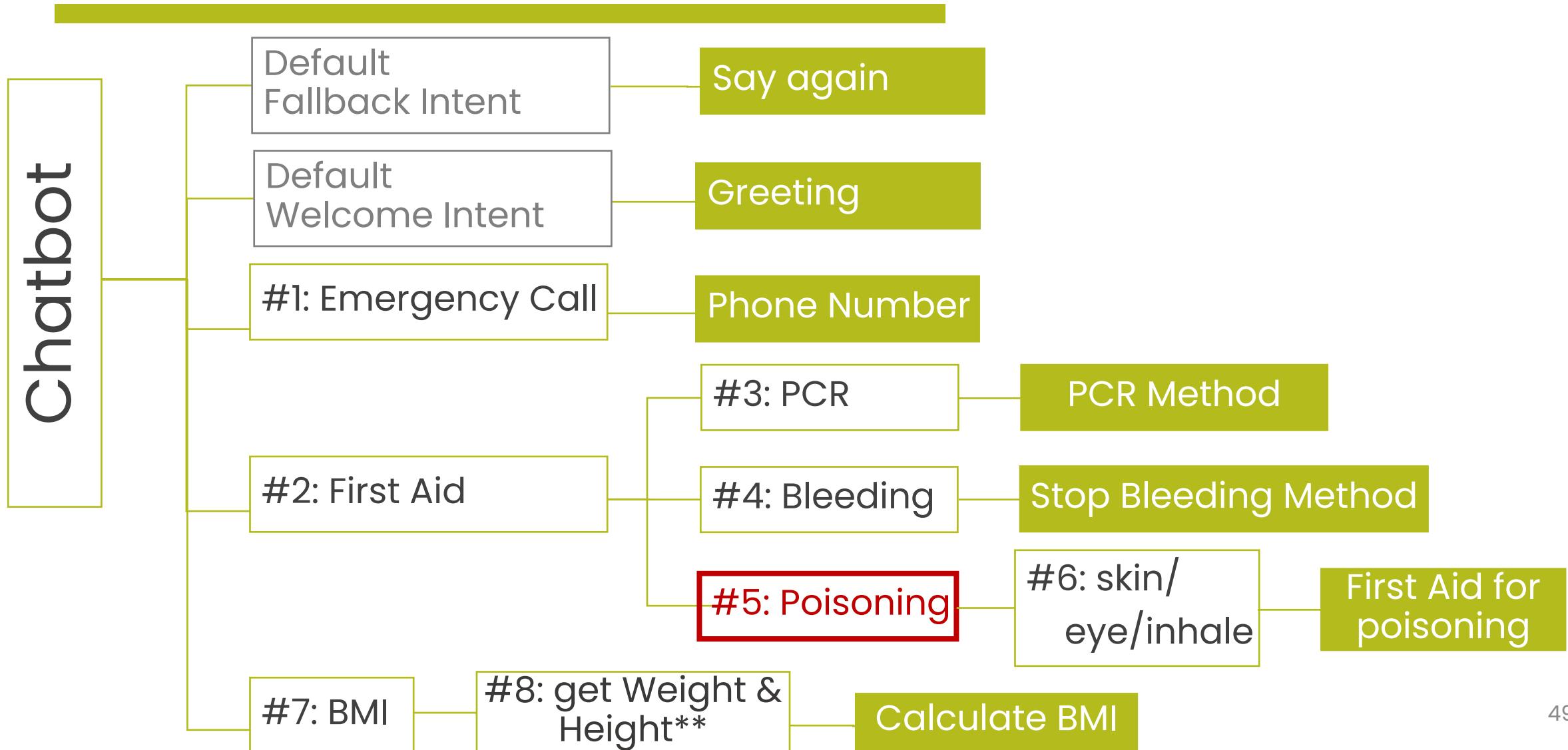
- Edit the card of intent#2 in Bleeding Section link to the youtube clip
E.g., <https://www.youtube.com/watch?v=NxO5LvgqZe0>



Intent#5

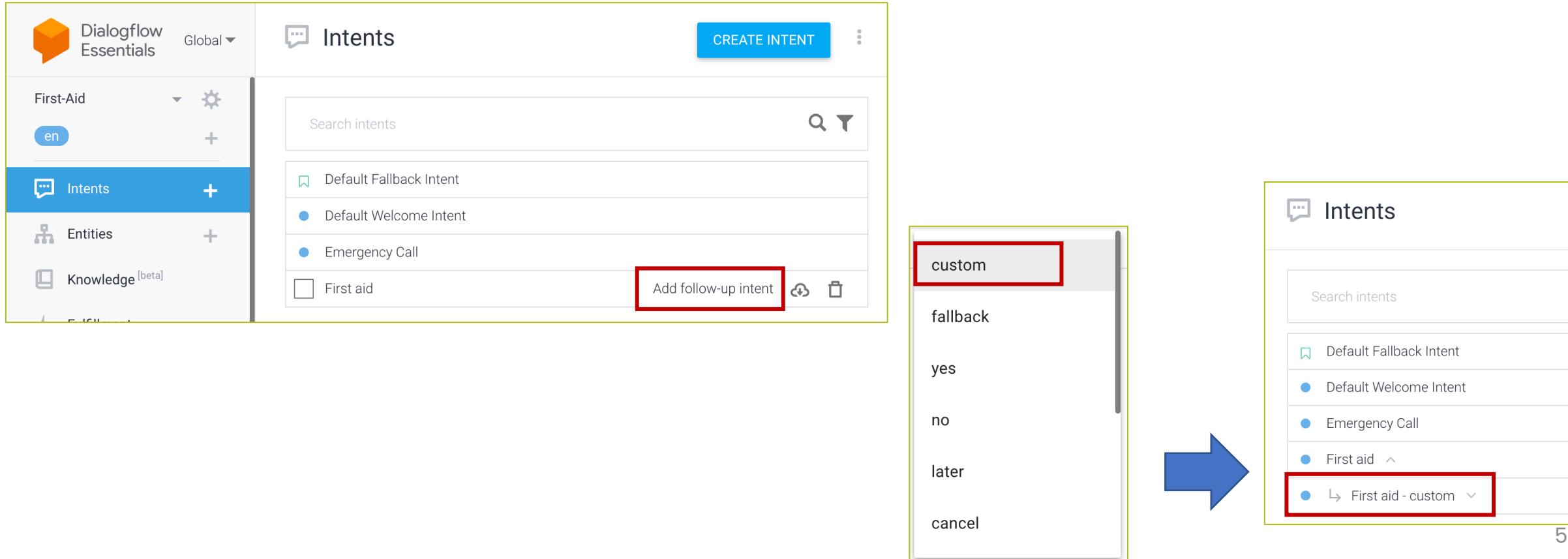
Poisoning

Intent Mapping



Intent#5

1. click “Add follow-up intent” after the First aid intent and select custom

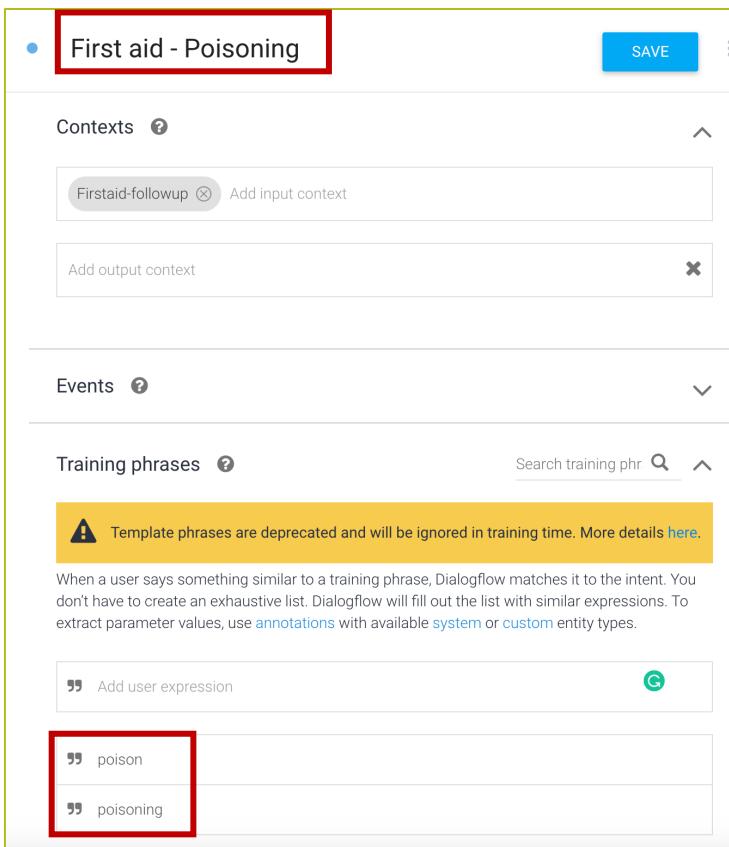


The screenshot illustrates the process of creating a custom follow-up intent in Dialogflow:

- Left Panel:** Shows the project "First-Aid" and language "en". The "Intents" section is selected, highlighted with a blue background.
- Intents List:** Displays several built-in intents: Default Fallback Intent, Default Welcome Intent, Emergency Call, and First aid. The "First aid" intent is currently selected, indicated by a red box around its name.
- Action Bar:** Contains a "CREATE INTENT" button and an "Add follow-up intent" button, which is also highlighted with a red box.
- Modal Window:** A modal window titled "custom" is open, listing five options: fallback, yes, no, later, and cancel. The word "custom" is also highlighted with a red box.
- Right Panel:** Shows the updated list of intents, where "First aid - custom" has been added as a new intent below the original "First aid" intent.
- Arrow:** A large blue arrow points from the "Intents" list on the left towards the "Intents" list on the right, indicating the progression of the action.

Intent#5

2. Rename to First aid - Poisoning
3. Add Training phrases



First aid - Poisoning

Contexts

Firstdaid-followup (X) Add input context

Add output context (X)

Events

Training phrases

A Template phrases are deprecated and will be ignored in training time. More details [here](#).

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use [annotations](#) with available [system](#) or [custom](#) entity types.

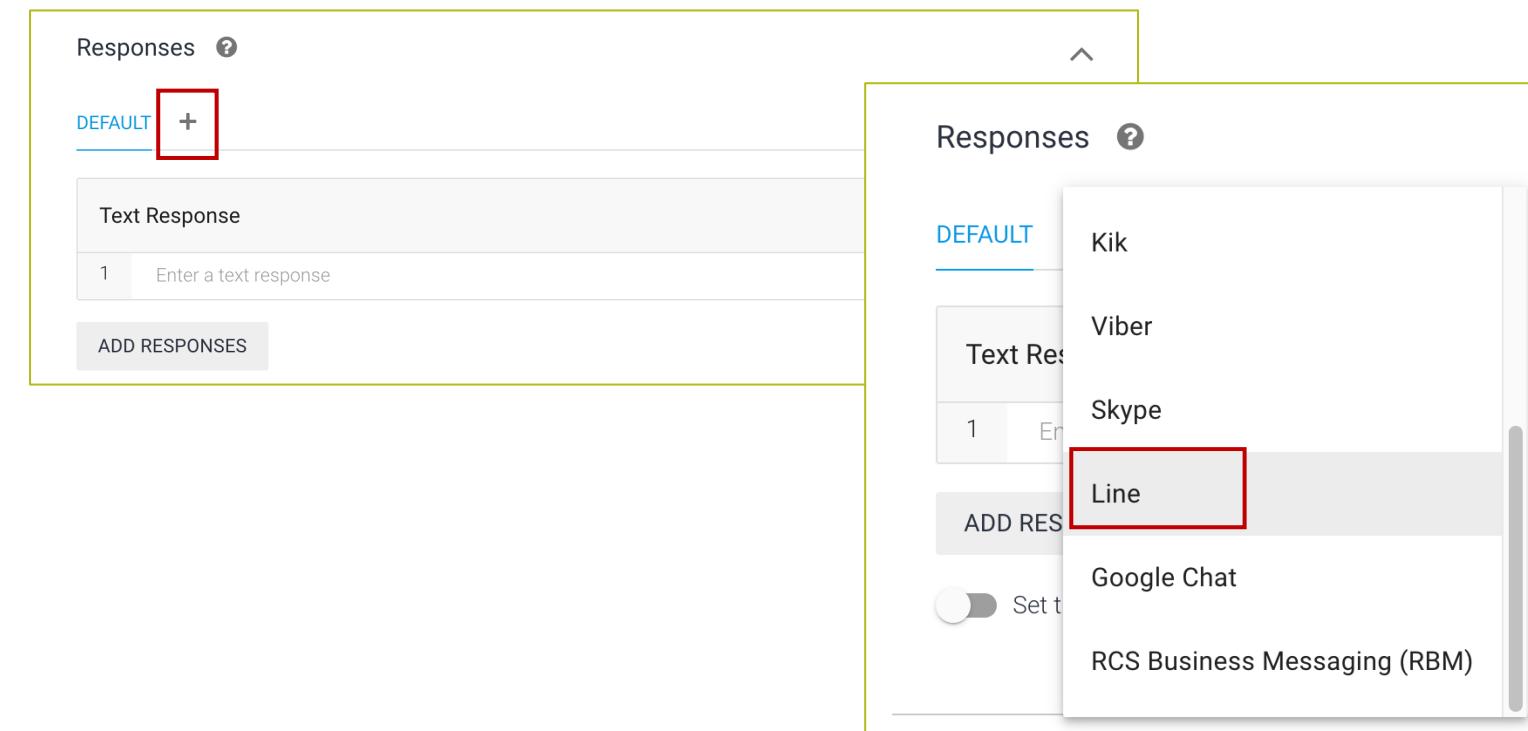
Add user expression (G)

poison

poisoning

SAVE

4. In Response Section click + and select LINE from the list



Responses

DEFAULT

+ (highlighted with a red box)

Text Response

1 Enter a text response

ADD RESPONSES

Responses

DEFAULT

Kik

Viber

Skype

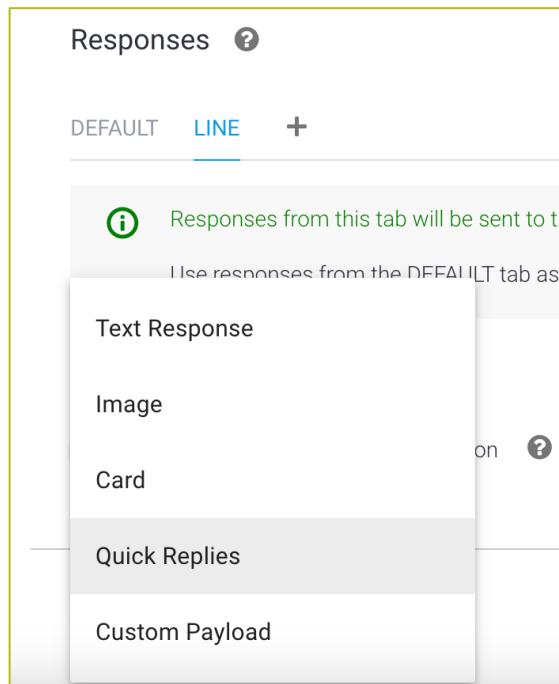
Line (highlighted with a red box)

Google Chat

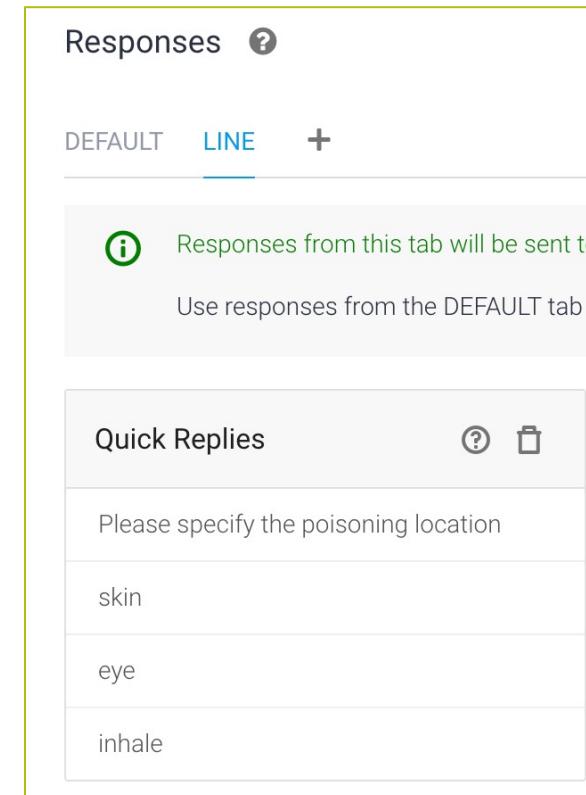
RCS Business Messaging (RBM)

Intent#5

5. Click Add Responses and Quick Replies



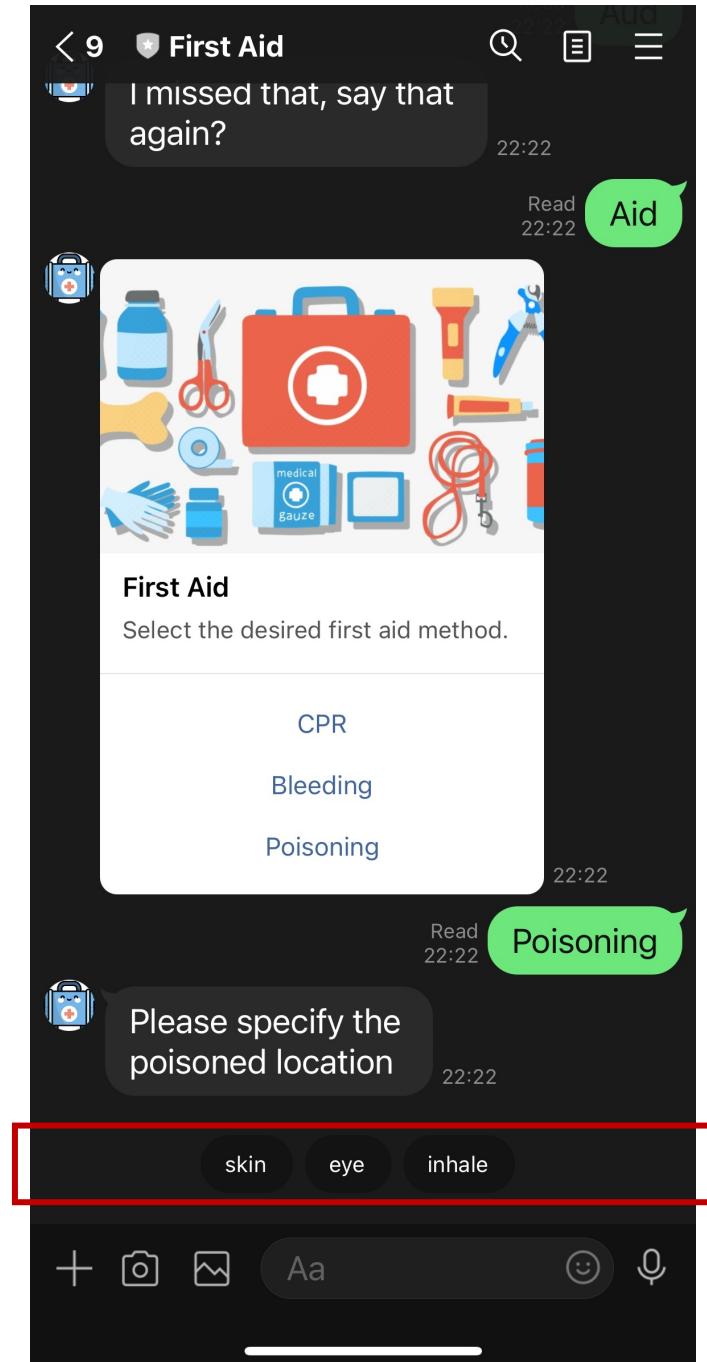
6. Input title and choices for the user



7. save the intent

Intent#5

8. Test with LINE Application in Mobile phone

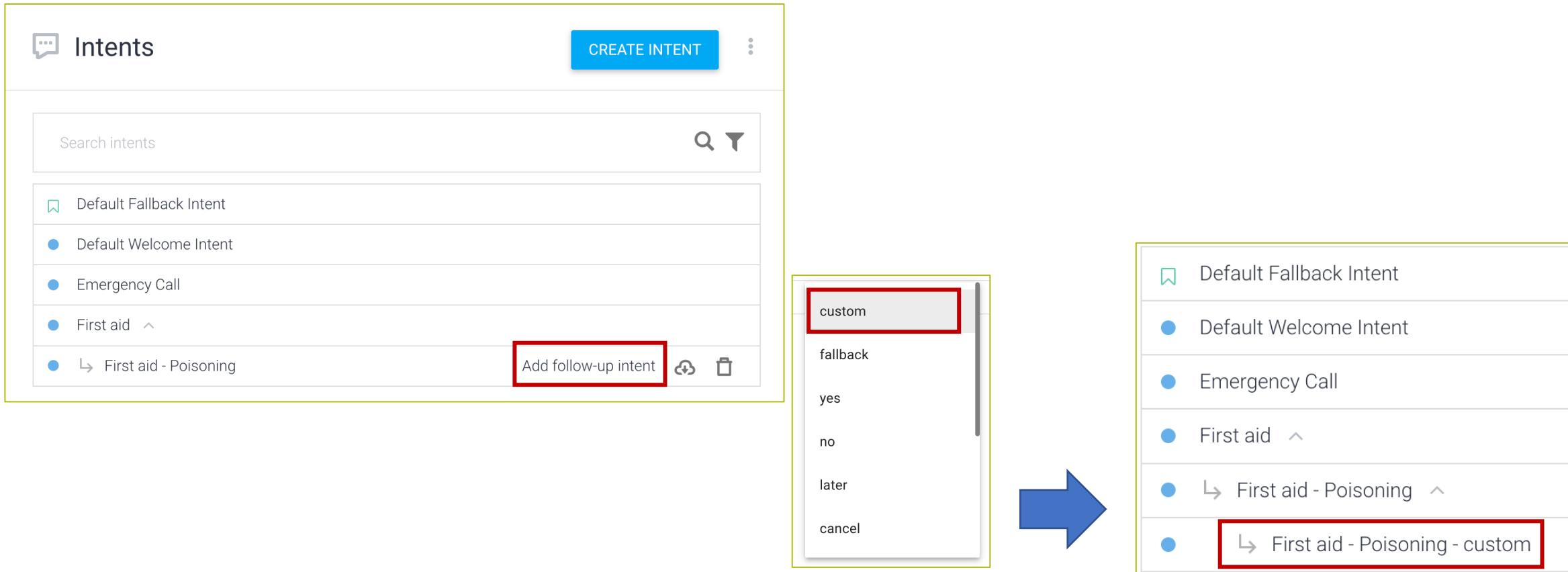


Intent#6

skin/eye/inhale

Intent#6

1. click “Add follow-up intent” after the First aid –Poisoning intent and select custom



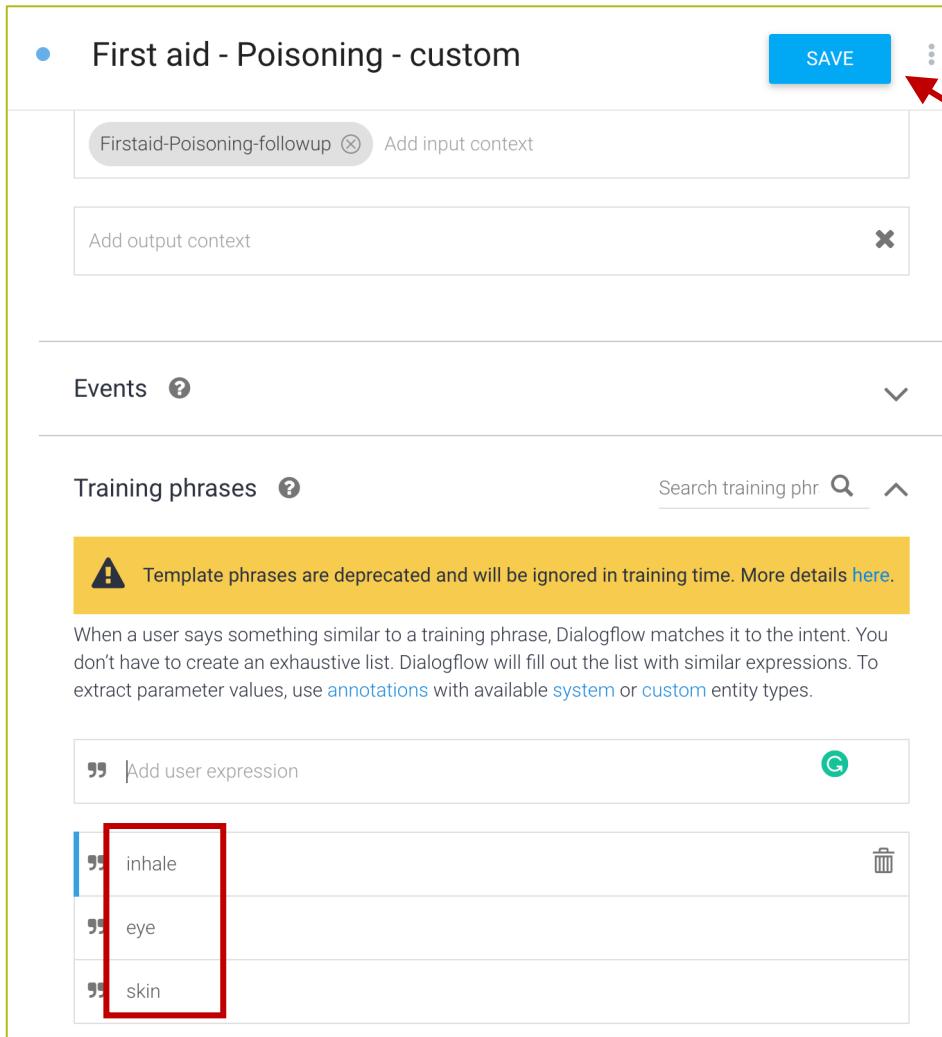
Intent#6

- First aid - Poisoning - custom

SAVE

⋮

2. Add Training phrases
3. Add Text Response
4. enable webhook in Fullfilment
5. Save the Intent



Firstaid-Poisoning-followup (x) Add input context

Add output context (x)

Events (?)

Training phrases (?)

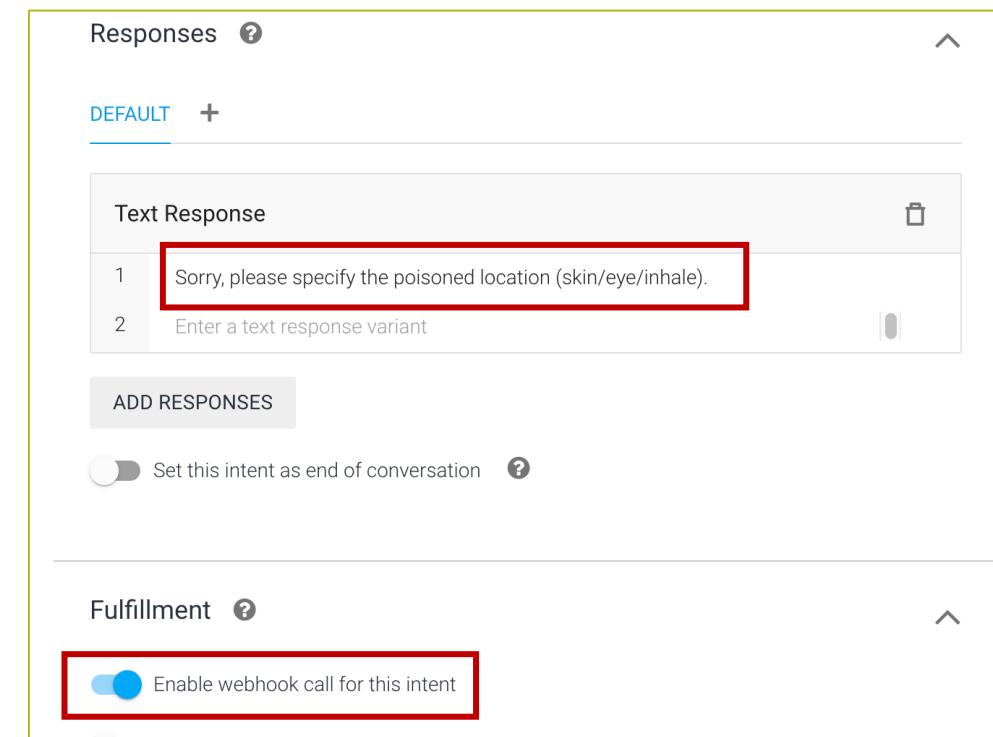
Search training phr (x) ^

⚠ Template phrases are deprecated and will be ignored in training time. More details [here](#).

When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use [annotations](#) with available [system](#) or [custom](#) entity types.

Add user expression (x)

inhale
eye
skin



Responses (?) ^

DEFAULT +

Text Response

1 Sorry, please specify the poisoned location (skin/eye/inhale).
2 Enter a text response variant

ADD RESPONSES

Set this intent as end of conversation (?)

Fulfillment (?) ^

Enable webhook call for this intent

Google Sheet

Google Sheet

- We will use the google sheet as a database for the query.
- <https://docs.google.com/spreadsheets/>
- Create the sheet contain the keyword and answer
e.g., https://docs.google.com/spreadsheets/d/1sxt3oFn2SxIEqWw-cnLNehAohgeX6QB8LhzmD98gn1g/edit?usp=sharing

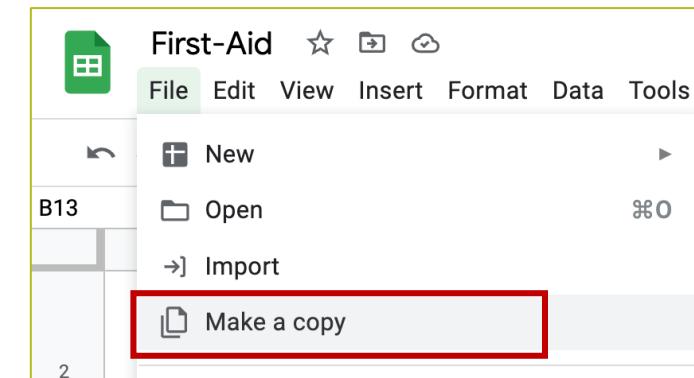
First-Aid   

File Edit View Insert Format Data Tools Extensions Help Last edit was 4 minutes ago

B16  |

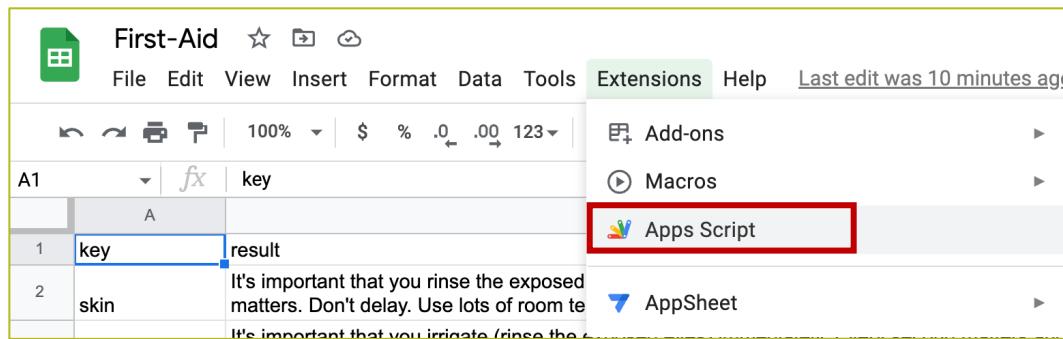
	A	B
1	key	result
2	skin	It's important that you rinse the exposed skin immediately. Remove contaminated clothing first (that's clothing with a spill). Every second matters. Don't delay. Use lots of room temperature running water and rinse for at least 15 minutes.
3	eye	It's important that you irrigate (rinse the exposed eyes) immediately. Every second matters and a delay could result in loss of sight. Remove contact lenses. Use lots of room temperature water and irrigate for at least 15 to 20 minutes.
4	inhaled	It's important that you move to fresh air immediately. Stay away from all toxic fumes and gases. Thoroughly ventilate the involved area.

Make a copy
to your Google Drive



App Script

1. Click menu Extensions and select App Script



2. Copy and paste the code in the area

```

var ss = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1SvbrT-LkG82DLc0_zKWv7XnUEDxiSTh4t0b6aeJfNSI/edit");
var sheet = ss.getSheetByName("Sheet1");

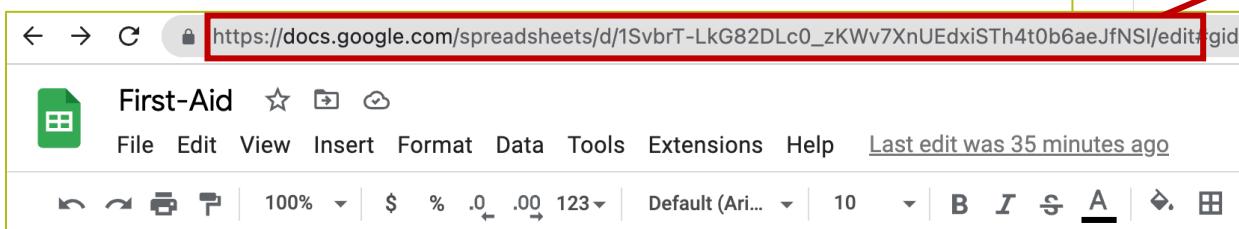
function doPost(e) {
  var data = JSON.parse(e.postData.contents);
  var userMsg = data.originalDetectIntentRequest.payload.data.message.text;
  var values = sheet.getRange(2, 1, sheet.getLastRow(), sheet.getLastColumn()).getValues();

  for(var i = 0;i<values.length; i++){
    if(values[i][0] == userMsg){
      i=i+2;
      var Data = sheet.getRange(i,2).getValue();

      var result = {
        "fulfillmentMessages": [
          {
            "platform": "line",
            "type": 4,
            "payload": {
              "line": {
                "type": "text",
                "text": Data
              }
            }
          }
        ]
      }
      var replyJSON = ContentService.createTextOutput(JSON.stringify(result)).setMimeType(ContentService.MimeType.JSON);
      return replyJSON;
    }
  }
}

```

3. Change the URL with your sheet URL



App Script

```
var ss = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/1SvbrT-LkG82DLc0_zKwv7XnUEdxisth4t0b6aeJfNSI/edit");
var sheet = ss.getSheetByName("sheet1");

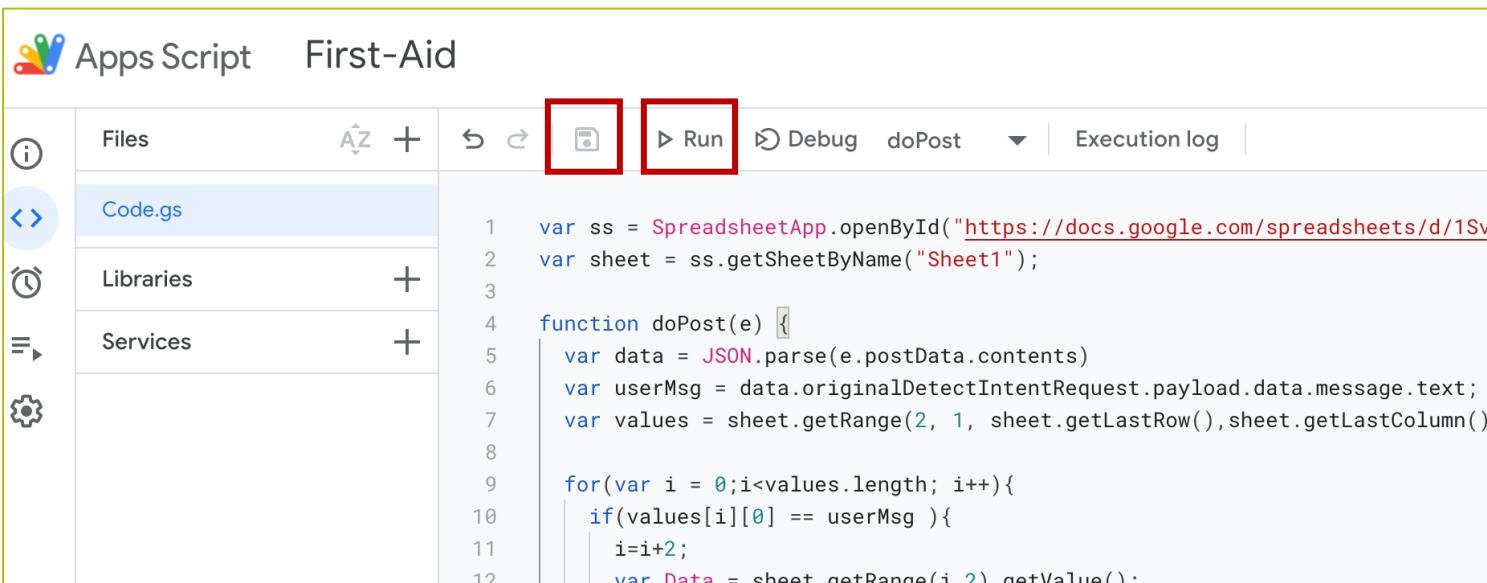
function doPost(e) {
  var data = JSON.parse(e.postData.contents)
  var userMsg = data.originalDetectIntentRequest.payload.data.message.text;

  var values = sheet.getRange(2, 1, sheet.getLastRow(), sheet.getLastColumn()).getValues();
  for(var i = 0;i<values.length; i++){
    if(values[i][0] == userMsg ){
      var Data = sheet.getRange(i+2,2).getValue();
      var result = {
        "fulfillmentMessages": [
          {
            "platform": "line",
            "type": 4,
            "payload" : {
              "line": {
                "type": "text",
                "text": Data
              }
            }
          }
        ]
      }
      var replyJSON = ContentService.createTextOutput(JSON.stringify(result)).setMimeType(ContentService.MimeType.JSON);
      return replyJSON;
    }
  }
}
```

App Script

4. Save the Script

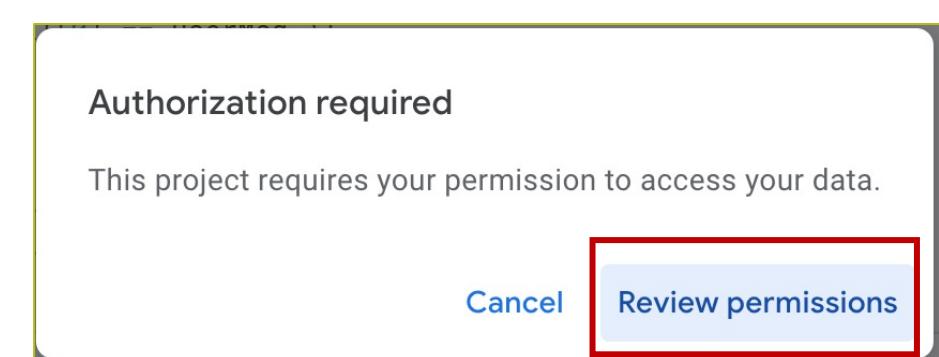
5. Run the Script



```

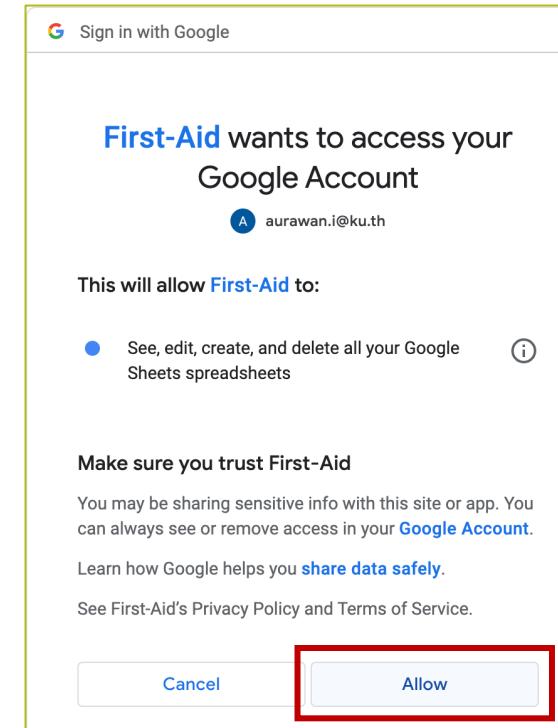
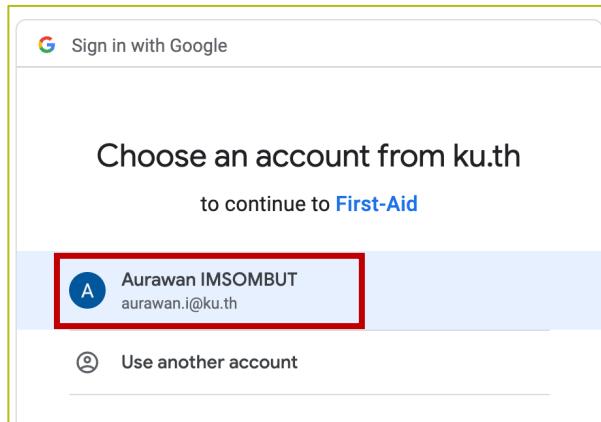
Apps Script First-Aid
Files A-Z + ↻ ⇢ Debug doPost Execution log
Code.gs
Libraries +
Services +
1 var ss = SpreadsheetApp.openById("https://docs.google.com/spreadsheets/d/1SvI");
2 var sheet = ss.getSheetByName("Sheet1");
3
4 function doPost(e) {
5   var data = JSON.parse(e.postData.contents)
6   var userMsg = data.originalDetectIntentRequest.payload.data.message.text;
7   var values = sheet.getRange(2, 1, sheet.getLastRow(), sheet.getLastColumn())
8
9   for(var i = 0;i<values.length; i++){
10     if(values[i][0] == userMsg ){
11       i=i+2;
12       var Data = sheet.getRange(i,2).getValue();
13     }
14   }
15   return ContentService.createTextOutput("Success")
16 }
  
```

6. Click Review permissions button



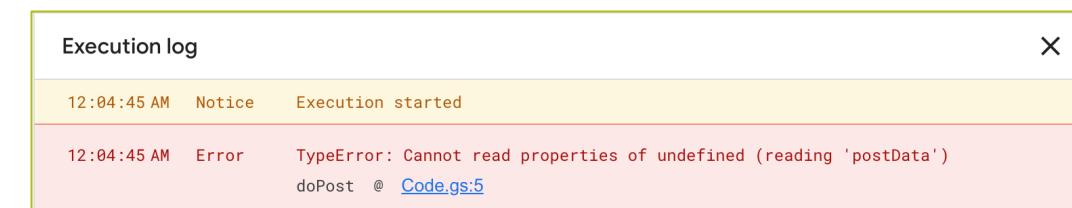
App Script

7. Sign in with your Google account



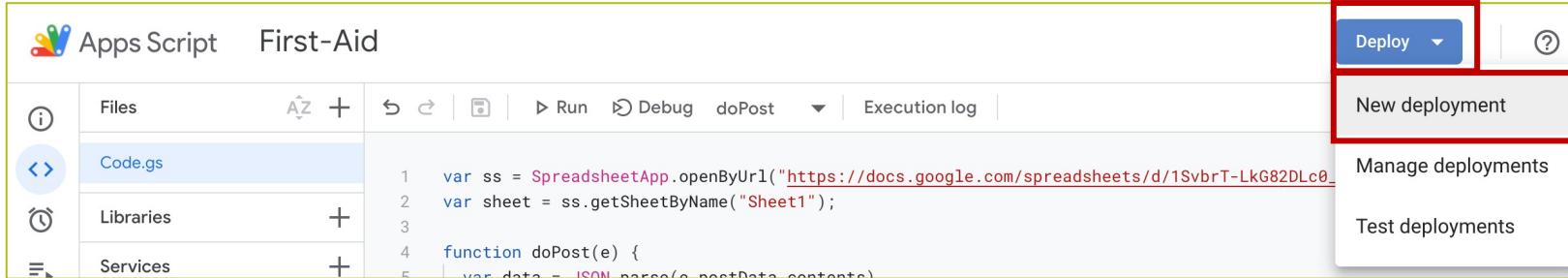
8. Press Allow button

the script will run and the error will show but it's OK in this case.

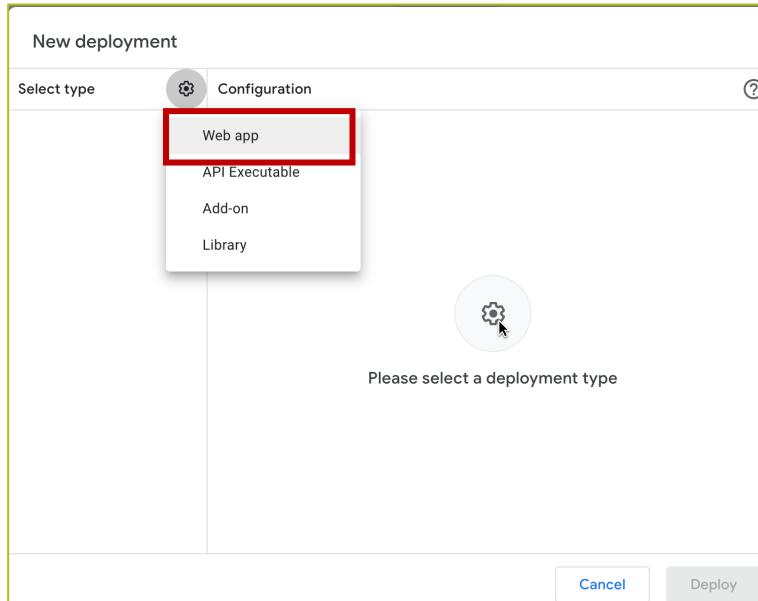


App Script

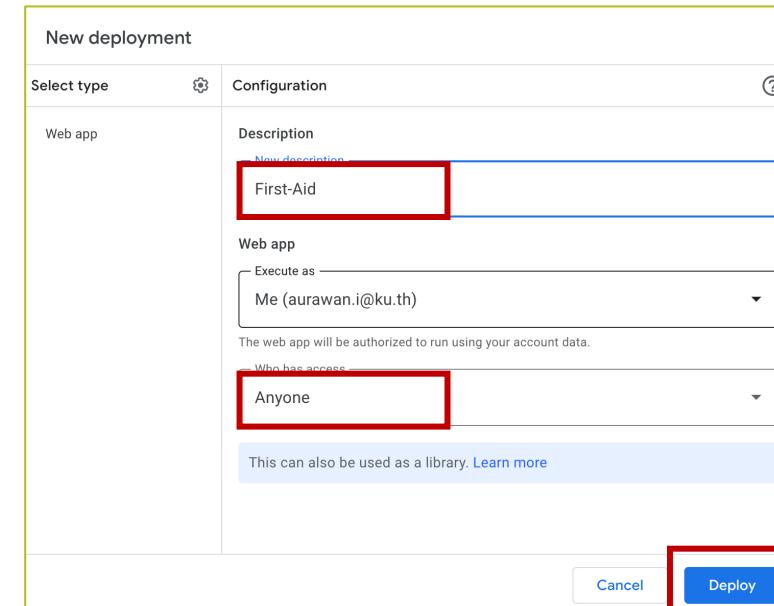
9. Click Deploy button > New deployment



10. Select deploy type as Web app



11. Input Name, Select Anyone and press Deploy button



App Script

12. Copy Web app URL for using in Dialogflow

New deployment

Deployment successfully updated.

Version 1 on Mar 9, 10:22 PM

Deployment ID

AKfycbyobr5T8wM1G7DG8U9ibt0ZYjFDgLQdLuvbRldz1kgAVSD14jeqUxaMHaKJmZh3KDe84g

 [Copy](#)

Web app

URL

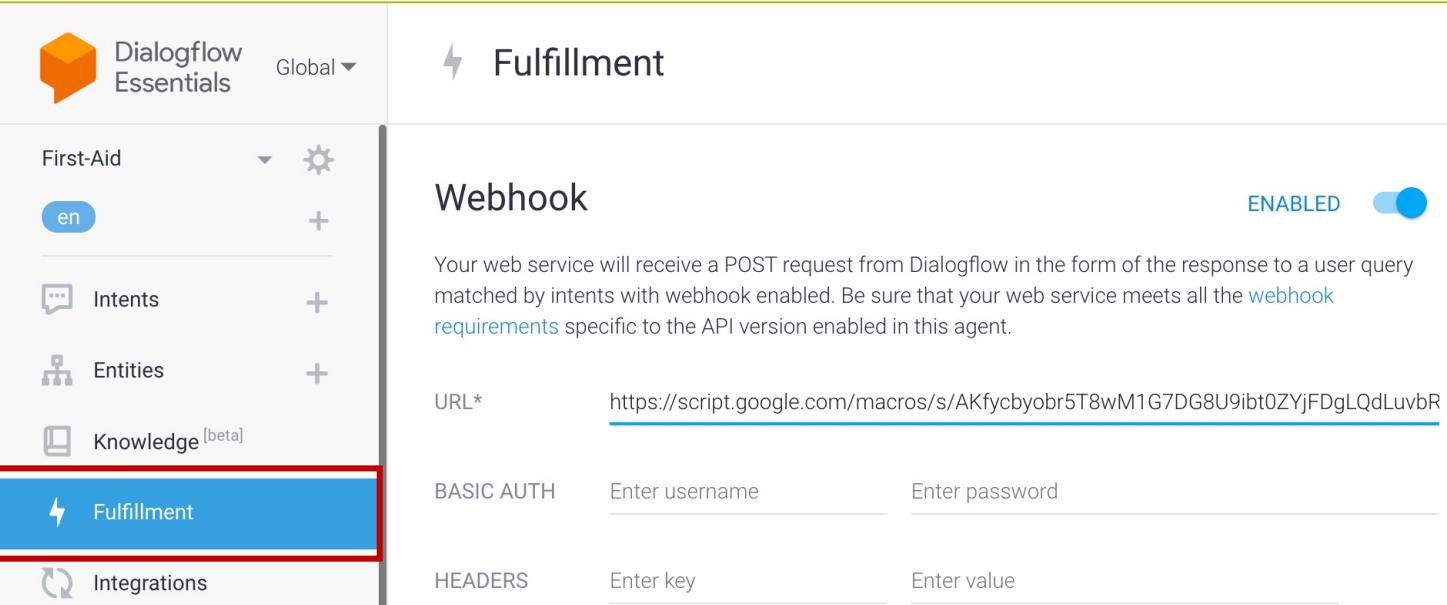
<https://script.google.com/macros/s/AKfycbyobr5T8wM1G7DG8U9ibt0ZYjFDgLQdLuvbRldz1kgAVSD14jeqUxaMHaKJmZ...>

 [Copy](#)

[Done](#)

Dialogflow

1. Select Fulfillment
2. Enabled Webhook
3. Paste the Web app URL from App Script



Fulfillment

Webhook

ENABLED

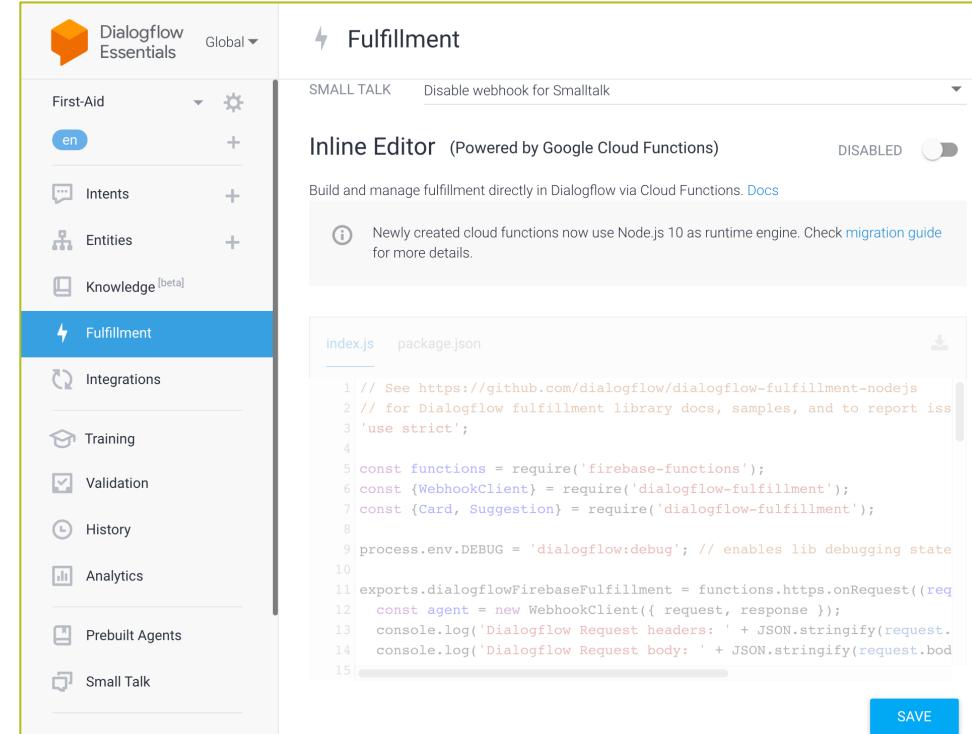
Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.

URL*

BASIC AUTH

HEADERS

4. Scroll down and click SAVE button



Fulfillment

SMALL TALK Disable webhook for Smalltalk

Inline Editor (Powered by Google Cloud Functions) DISABLED

Newly created cloud functions now use Node.js 10 as runtime engine. Check [migration guide](#) for more details.

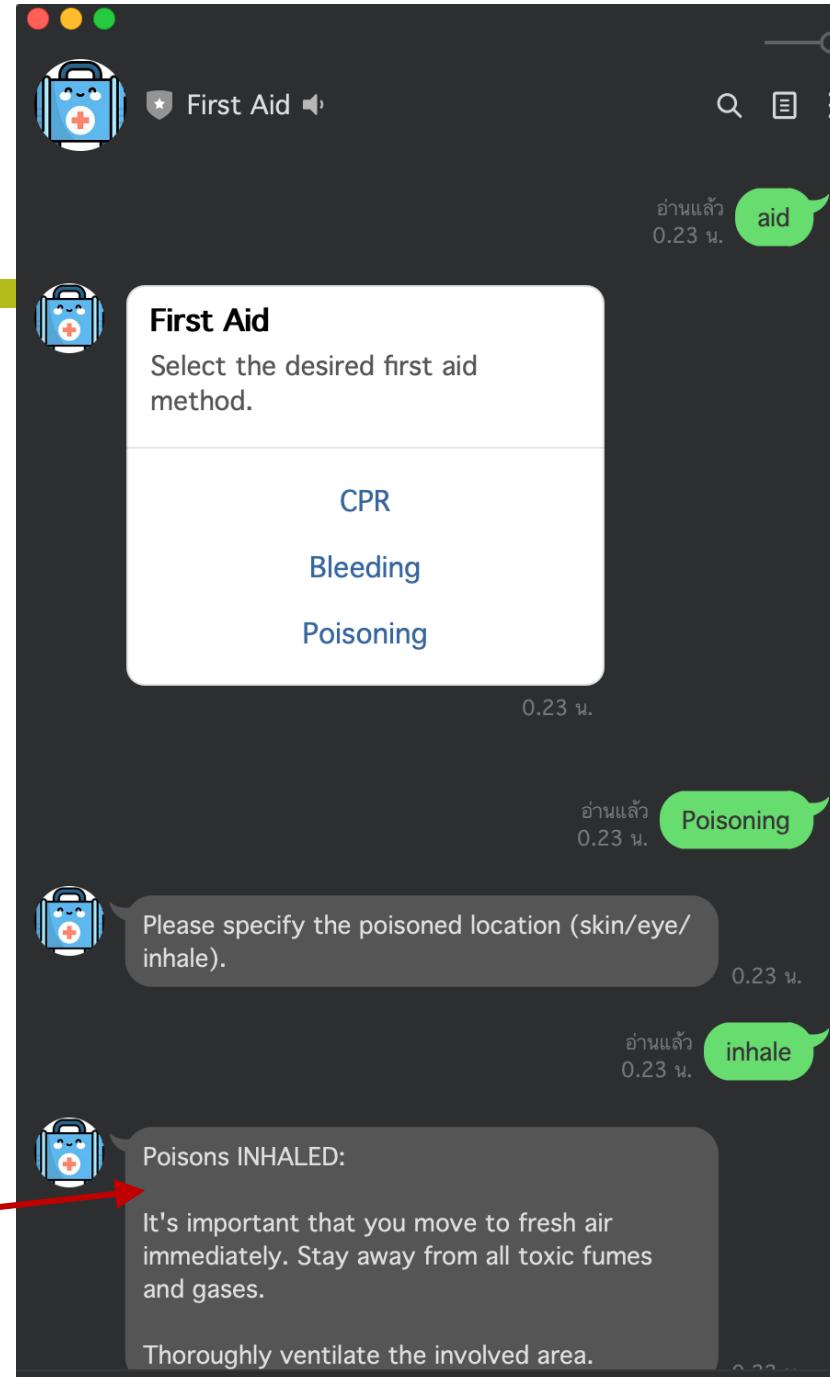
```

index.js package.json
1 // See https://github.com/dialogflow/dialogflow-fulfillment-nodejs
2 // for Dialogflow fulfillment library docs, samples, and to report issues
3 'use strict';
4
5 const functions = require('firebase-functions');
6 const {WebhookClient} = require('dialogflow-fulfillment');
7 const {Card, Suggestion} = require('dialogflow-fulfillment');
8
9 process.env.DEBUG = 'dialogflow:debug'; // enables lib debugging state
10
11 exports.dialogflowFirebaseFulfillment = functions.https.onRequest((req,
12   const agent = new WebhookClient({ request, response });
13   console.log('Dialogflow Request headers: ' + JSON.stringify(request.headers));
14   console.log('Dialogflow Request body: ' + JSON.stringify(req.body));
15
16   return agent.handleRequest();
17 });
  
```

SAVE

Intent#6

- Test with LINE Application

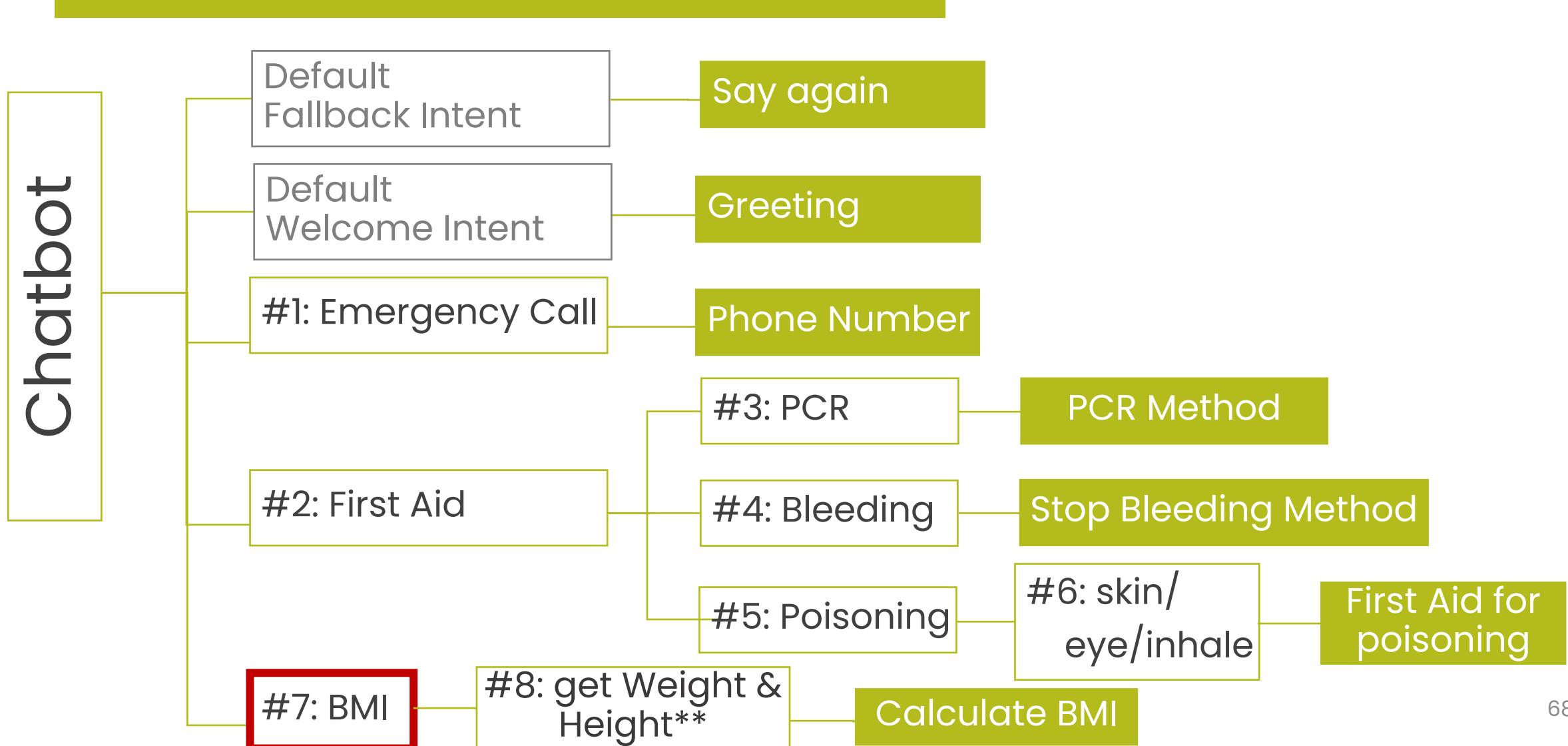


Data from Google Sheet

Intent#7 & 8

BMI

Intent Mapping



Intent#7

1. create intent bmi

The screenshot shows the 'bmi' intent configuration page. At the top, there's a blue 'SAVE' button and a three-dot menu icon. Below it, sections for 'Contexts' and 'Events' have dropdown arrows. The 'Training phrases' section contains a yellow warning box stating: 'Template phrases are deprecated and will be ignored in training time. More details [here](#)'. It also includes a 'Search training ph' input field and a 'Add user expression' button with a microphone icon. A list box at the bottom contains the phrase 'bmi'. The entire interface is framed by a thick green border.

The screenshot shows the 'Responses' configuration for the 'bmi' intent. It starts with a 'DEFAULT' section with a '+' button. Below it is a 'Text Response' section containing two variants: '1 please enter your weight (kg.) and height (cm.)' and '2 Enter a text response variant'. There's a 'Text Response' button with a trash icon next to each variant. At the bottom, there's an 'ADD RESPONSES' button and a toggle switch labeled 'Set this intent as end of conversation' with a question mark icon.

Intent#8

2. Add follow-up custom intent of bmi with bmi – custom intent

The screenshot shows the 'Intent' section of the Dialogflow interface. A dropdown menu is open, showing 'bmi' and 'bmi - custom'. 'bmi - custom' is currently selected, indicated by a blue dot and a downward arrow.

3. Add training phrases

The screenshot shows the 'Training phrases' section for the 'bmi - custom' intent. It includes a warning message about deprecated template phrases and a list of user expressions:

- weight 60 height 170
- 55, 160
- 50 kg, 165 cm

The screenshot shows the 'Action and parameters' and 'Responses' sections of the intent configuration. In the 'Action and parameters' section, two parameters are defined:

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST	PROMPTS
<input checked="" type="checkbox"/>	weight	@sys.number	\$weight	<input type="checkbox"/>	Please enter yo... Please enter yo...
<input checked="" type="checkbox"/>	height	@sys.number	\$height	<input type="checkbox"/>	Please enter yo... Please enter yo...
<input type="checkbox"/>	Enter nai	Enter ent	Enter value	<input type="checkbox"/>	

In the 'Responses' section, there is a single text response:

1 Your weight is \$weight and your height is \$height

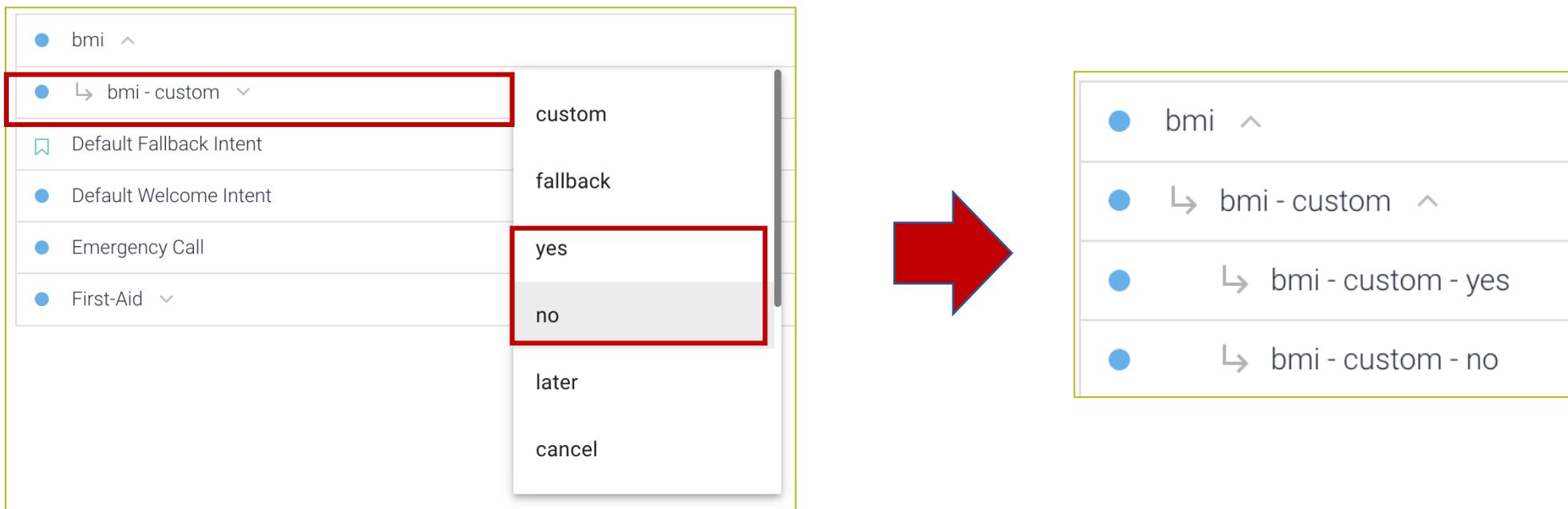
4. Add parameters and response text

5. Tag parameter in training phrases

6. Save

Intent#9, 10

7. Add follow-up yes intent of bmi - custom intent
8. Add follow-up no intent of bmi - custom intent



Intent#9

9. Add output context and text response of bmi – custom – no

The screenshot shows the configuration of an intent named "bmi - custom - no".

Intent Configuration:

- Intent Name: bmi - custom - no
- Contexts:
 - bmi-custom-followup (Add input context)
 - 5 bmi-followup (Add output context)
- Buttons: SAVE and More options (three dots)

Responses:

- DEFAULT +
- Text Response
 - 1 Please enter your weight and height again
 - 2 Enter a text response variant

Intent#10

10. Add Action and parameters, Test responses and enable fulfillment of bmi -custom -yes

- bmi - custom - yes

SAVE ⋮

Action and parameters

bmi.bmi-custom.bmi-custom-yes

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	weight	Enter entity	#bmi-custom-followup.weight	<input type="checkbox"/>
<input type="checkbox"/>	height	Enter entity	#bmi-custom-followup.height	<input type="checkbox"/>
<input type="checkbox"/>	bmi	Enter entity	bmi	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

Responses ?

DEFAULT +

Text Response

1 please wait a moment

2 Enter a text response variant ⋮

ADD RESPONSES

Set this intent as end of conversation ?

Fulfillment ?

Enable webhook call for this intent

Enable webhook call for slot filling

Intent#10

11. Edit Apps Script for calculate BMI

```
1 var ss = SpreadsheetApp.openByUrl("https://docs.google.com/spreadsheets/d/  
1SvbrT-LkG82DLc0_zKWv7XnUEdxISTh4t0b6aeJfNSI/edit");  
2 var sheet = ss.getSheetByName("sheet1");  
3  
4 function doPost(e) {  
5   var data = JSON.parse(e.postData.contents)  
6   var userMsg = data.originalDetectIntentRequest.payload.data.message.text;  
7   var bmi = data.queryResult.parameters.bmi;  
8   if (bmi != 'bmi') {  
9     var values = sheet.getRange(2, 1, sheet.getLastRow(), sheet.getLastColumn()).getValues();  
10    > for(var i = 0;i<values.length; i++){...  
31    }  
32  }
```

```
33 else{  
34   var weight = data.queryResult.parameters.weight;  
35   var height = data.queryResult.parameters.height/100;  
36   var bmi_value = (weight/(height*height)).toFixed(2);  
37   if(bmi<18.5){  
38     text = 'ค่า BMI=' + bmi_value + ', ผอมเกินไป ควรทานอาหารให้เยอะขึ้น';  
39   }  
40   else if(bmi < 24){  
41     text = 'ค่า BMI=' + bmi_value + ', หุ่นดี';  
42   }  
43   else if(bmi < 30){  
44     text = 'ค่า BMI=' + bmi_value + ', อ้วน';  
45   }  
46   else{  
47     text = 'ค่า BMI=' + bmi_value + ', อ้วนมาก';  
48   }  
49   var result = {  
50     "fulfillmentMessages": [  
51       {  
52         "platform": "line",  
53         "type": 4,  
54         "payload" : {  
55           "line": {  
56             "type": "text",  
57             "text": text  
58           }  
59         }  
60       }  
61     ]  
62   }  
63   var replyJSON = ContentService.createTextOutput(JSON.stringify(result)).setMimeType(ContentService.  
MimeType.JSON);  
64   return replyJSON;  
65 }  
66 }
```

Intent#10

12. Deploy Apps script and copy Web App URL for updating in fulfillment of Dialogflow

New deployment

Deployment successfully updated.

Version 13 on Mar 19, 11:58 PM

Deployment ID
AKfycbwufnwQ2ukYNjBCFTTYTThELewrGRQOoFLs0UIZL6e9SfnVb90_ffs704eMfzStzaSPLg

Copy

Web app

URL
https://script.google.com/macros/s/AKfycbwufnwQ2ukYNjBCFTTYTThELewrGRQOoFLs0UIZL6e9SfnVb90_ffs704eMfzs...

Copy

Done

Dialogflow Essentials Global ▾

First-Aid en +

Intents +

Entities +

Knowledge [beta]

Fulfillment

Webhook ENABLED

Your web service will receive a POST request from Dialogflow in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the [webhook requirements](#) specific to the API version enabled in this agent.

URL*

BASIC AUTH Enter username Enter password

HEADERS Enter key Enter value

[+ Add header](#)

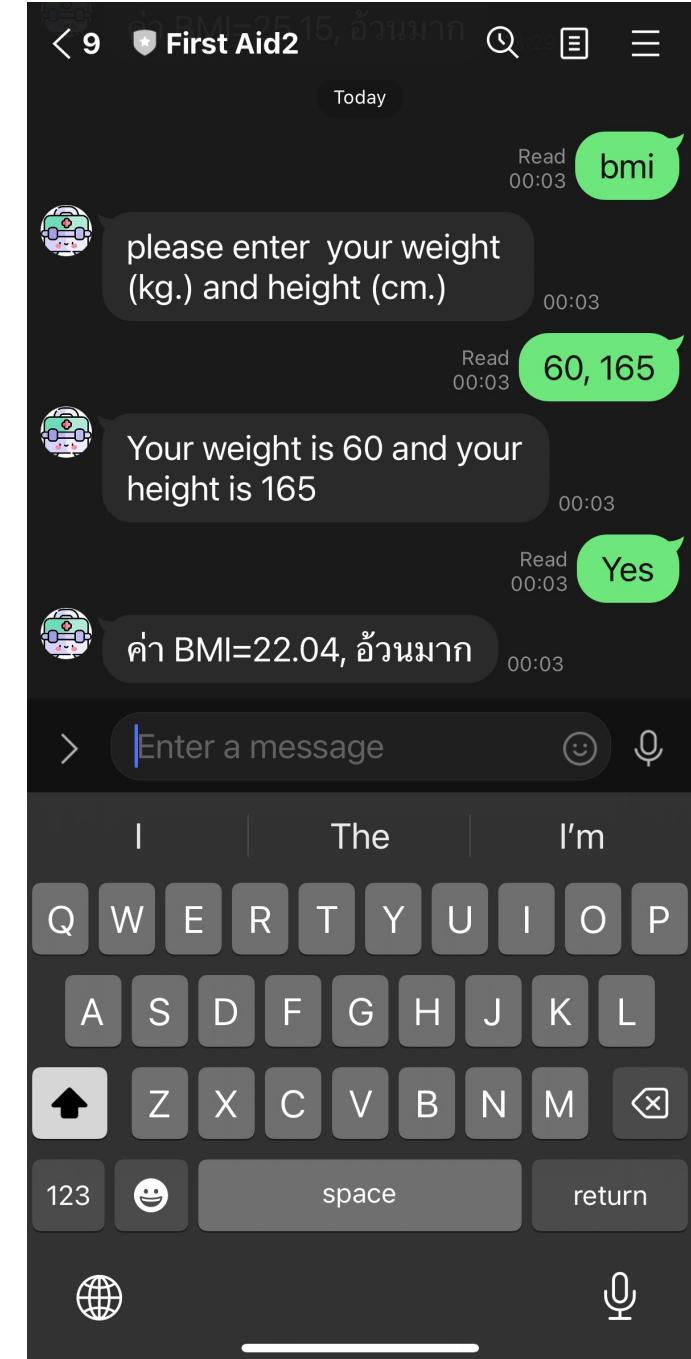
SMALL TALK Disable webhook for Smalltalk

13. Save

SAVE

Intent#7-10

- Test with LINE Application



Reference

- <https://developers.line.biz/en/reference/messaging-api/>
- <https://cloud.google.com/dialogflow>
- <https://cloud.google.com/dialogflow/es/docs>
- <https://medium.com/@jirawatee>