



# iNeuron

## CHATBOT HANDBOOK

## What is Chatbot?

A chatbot is a computer program that uses artificial intelligence to replicate human conversations. They're also referred to as "digital assistants" that are aware of human capabilities. Bots decipher the user's purpose, process their requests, and respond quickly with appropriate information.

Bots can speak via voice and text, and they can be used on websites, apps, and messaging platforms like Facebook Messenger, Twitter, and WhatsApp.

## History of Chatbot

In the 1960s, MIT professor Joseph Weizenbaum created the first chatbot. ELIZA was the name of the chatbot. Later on, you'll learn more about ELIZA and other popular chatbots built in the second half of the twentieth century.

In 2009, a Chinese firm called WeChat developed a more powerful Chatbot. WeChat has won the hearts of many users who have shown steadfast allegiance to it since its introduction. It's a really popular social media platform.

It has made it possible to construct extremely basic chatbots using their platform. It has become one of the most popular techniques for marketers and companies to lessen the amount of work they have to perform while interacting with clients online.

Though it has drawbacks and is less performant than today's chat platforms like Facebook Messenger, Slack, and Telegram, WeChat can still be used to build a sophisticated bot. Chumen Wenwen Company, which was created in 2012 by a former Google employee, has developed a powerful WeChat bot.

Early in 2016, the first wave of artificial data technology was included into the construction of chatbots. Customers could carry out parts of their regular activities from within their messaging platform, thanks to social media sites like Facebook, which allowed developers to create a chatbot for their product or service.

The arrival of chatbots in a community has ushered in the era of the conversational interface. It's an interface that won't require a screen or a mouse in the near future. The user interface will be completely conversational, and the communications will be indistinguishable from those we have with our friends and family.

## ELIZA

As previously stated, ELIZA was the very first chatbot. It was designed by Joseph Weizenbaum in 1966 and simulates conversation using pattern matching and replacement methods.

Welcome to

EEEEEE	LL	III	ZZZZZ	AAAAA
EE	LL	II	ZZ	AA AA
EEEEEE	LL	II	ZZZ	AAAAAAA
EE	LL	II	ZZ	AA AA
EEEEEE	LLLLL	III	ZZZZZ	AA AA

Eliza is a mock Rogerian psychotherapist.  
The original program was described by Joseph Weizenbaum in 1966.  
This implementation by Norbert Landsteiner 2005.

ELIZA: Is something troubling you ?  
YOU: Men are all alike.  
ELIZA: What is the connection, do you suppose ?  
YOU: They're always bugging us about something or other.  
ELIZA: Can you think of a specific example ?  
YOU: Well, my boyfriend made me come here.  
ELIZA: Is it important to you that your boyfriend made you come here ?  
YOU: He says I'm depressed much of the time.  
ELIZA: I am sorry to hear that you are depressed.  
YOU: It's true. I am unhappy.  
ELIZA: Can you explain what made you unhappy ?  
YOU:

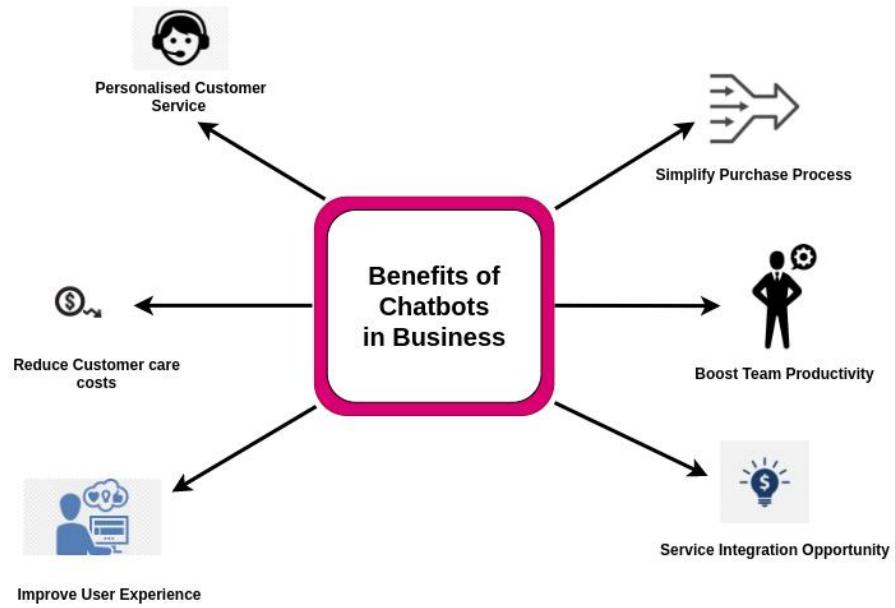
## Why Chatbot?

To begin with, texting has evolved into a natural mode of communication. It's as natural as breathing to use texting apps.

WhatsApp is used by over 2 billion people! around the world. Users of Facebook Messenger are expected to reach a similar amount in the near future. Chatbots are available in both of these apps.

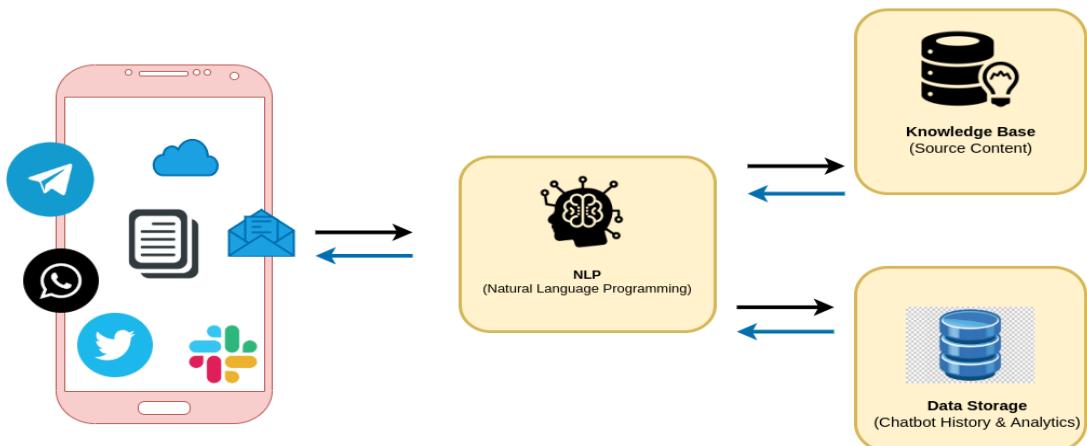
To automate discussions, you can use a chatbot for WhatsApp or Facebook. If your mother is the only one who texts you, this may not seem like a big concern. Chatbots, on the other hand, are invaluable for any type of business. It's like having a free secretary available 24 hours a day, 7 days a week!

Another significant benefit of chatbots is that they become attached to their users. A chatbot conversation is a particularly personal type of engagement. Some people consider chatbots to be their buddies who helped them survive the coronavirus outbreak.



## How Chatbot Works?

The most crucial duty of a chatbot is to analyze and detect the purpose of the user's request in order to extract relevant entities. The user receives an appropriate response after the analysis is completed.



Three classification methods are used by the chatbots.

### **Pattern matching**

Pattern matching are used by bots to group text, and this results in an appropriate response from clients. AIML (Artificial Intelligence Markup Language) is a structured model of these patterns that is widely used. In the connected pattern, a bot can acquire the correct answer. Anything linked to the connected patterns triggers a response from the bots.

### **Natural language understanding (NLU)**

Natural language understanding (NLU) refers to a chatbot's capacity to comprehend human speech. It is the process of transforming unstructured text into structured data that can be understood by a machine. NLU is based on three distinct ideas. Entities, context, and expectations are the three.

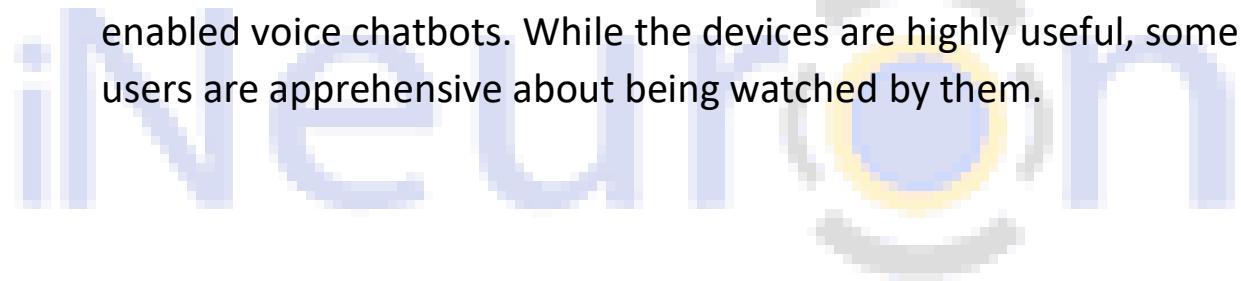
### **Natural language processing (NLP)**

Bots that use Natural Language Processing (NLP) are meant to turn a user's text or speech into structured data. The information is then used to select an appropriate response. Tokenization, chatbot sentiment analysis, entity recognition, and dependency parsing are all crucial steps in NLP.

## **Types of Chatbot**

Chatbots can be categorized into three categories.

1. Website Chatbot: It may be opened in your browser and used just like a regular website. They can sometimes appear as a little widget. They can also be used as a distinct landing page. Some chatbot editors allow you to construct a chatbot for a website as well as a chatbot website.
2. Messenger Chatbot: On social media and instant messaging services, chatbots are extremely popular. They're frequently more complex to create and come with a lot of restrictions. A Facebook chatbot, on the other hand, can be a fantastic complement to a business page.
3. Voice Chatbot: Millions of people use Alexa or Siri from Amazon without considering them to be chatbots. These virtual assistants, on the other hand, are exactly that. They're artificial intelligence-enabled voice chatbots. While the devices are highly useful, some users are apprehensive about being watched by them.



## Google Dialogflow Chatbot

**Step 1.** Login to [Dialogflow](#)



You need to login first with your Google account (if you don't have an account in Google, you can [sign up here](#))

## Step2. Create a new agent

The screenshot shows the Dialogflow Essentials interface. On the left, there's a sidebar with a 'Table\_Booking' project selected. The sidebar has a 'Create new agent' button highlighted with a red box. The main panel is titled 'Agents' and shows a table with one row for 'Table\_Booking'. The table includes columns for 'en' and 'Search agents'. A search bar and a magnifying glass icon are also present. In the top right corner of the main panel, there's a blue 'CREATE AGENT' button, which is also highlighted with a red box. Red arrows point from the text labels to their respective highlighted buttons.

Follow the arrow, 'Create Agent' button is there, select it.

After Selecting the 'Create Agent':

iNeuron\_Restaurant

DEFAULT LANGUAGE ⓘ

English – en

Primary language for your agent. Other languages can be added later.

DEFAULT TIME ZONE

(GMT+6:00) Asia/Almaty

Date and time requests are resolved using this timezone if not provided in the API requests.

GOOGLE PROJECT

Create a new Google project

Enables Cloud functions, Actions on Google and permissions management.

AGENT TYPE

Set as Mega Agent

Combine multiple Dialogflow agents (i.e. sub agents) into a single agent (i.e. [mega agent](#)).

1. We have given our agent's name or bot name as 'iNeuron\_Restaurant'
2. Then, click on 'CREATE'. Your agent would be created.

(Keep the language and Time zone default no need to do change in there.)

### Step3. Create an Intent

Dialogflow Essentials Global ▾

iNeuron\_Restaurant en

**Intents**

Search intents

Default Fallback Intent

Default Welcome Intent

CREATE INTENT

No regular intents yet. [Create the first one.](#)

Intents are mappings between a user's queries and actions fulfilled by your software. [Read more here.](#)

Before you start, check out [Prebuilt Agents](#), a collection of agents developed by the Dialogflow team.

1. You can see here, our bot\_name is reflected.
2. With these buttons we can create our intent, ‘Default Welcome Intent’ have greetings intents are available, we will change intent for our restaurant bot.

Intents

CREATE INTENT

Search intents

Default Fallback Intent

Default Welcome Intent

We're going to edit ‘Default Welcome Intent’ for our restaurant chatbot. Click on the ‘Default Welcome Intent’, you’ll land into the below page.

- Default Welcome Intent

Contexts

Events

Welcome Add event

Training phrases

Search training phrases

Here we will delete the default responses and add our responses inside that.

Responses

DEFAULT

Text Response



- 1 Hi! How are you doing?
- 2 Hello! How can I help you?
- 3 Good day! What can I do for you today?
- 4 Greetings! How can I assist?
- 5 Enter a text response variant

After, deletion will have to click on 'ADD RESPONSES' and select 'Text Response'.

Responses

DEFAULT

Text Response

Custom Payload

You can check here; we changed our responses.

Responses

DEFAULT

## Text Response

- 1 Hello, My name is Hodor. I am the Virtual Assistant of iNeuron Restaurant. Would you like to order food or looking for a table.
- 2 Hello! My name is Hodor. How may I help you?
- 3 Hello! My name is Hodor. Am a Virtual Assistant of iNeuron Restaurant. How can i help you?
- 4 Enter a text response variant



We have to save our intents with the button ‘SAVE’ then your intent would be saved.

- Default Welcome Intent

Contexts



Events



Welcome Add event

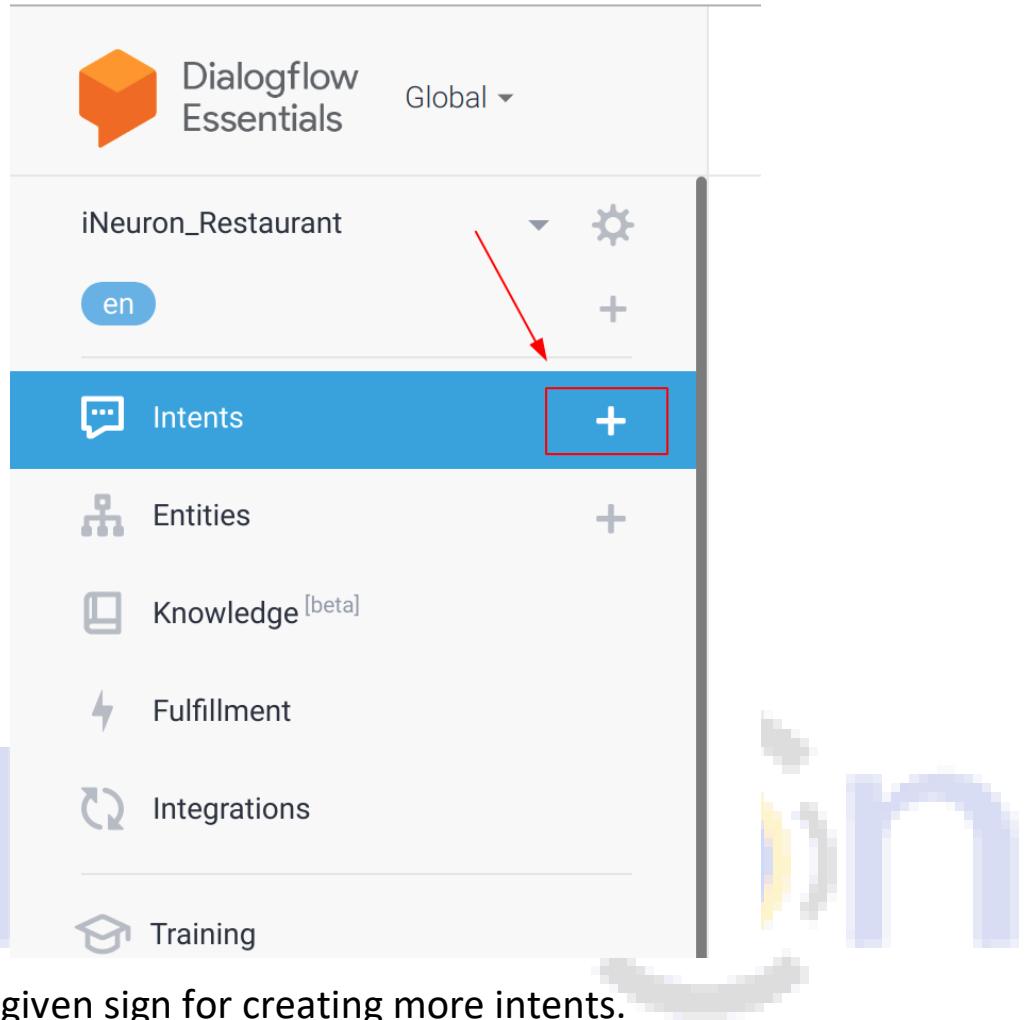
Now, will check on the right-hand side our intent is working or not.

The screenshot shows a user interface for a virtual assistant. At the top, there's a search bar with the placeholder "Try it now" and a microphone icon. Below it, the word "Agent" is written in blue. A red arrow points from the "USER SAYS" section to a red circle containing a question mark icon. The "USER SAYS" section contains the text "Hello". To the right of the user input, there's a "COPY CURL" button with a red arrow pointing to it. Below the user input, there's a "DEFAULT RESPONSE" section with a dropdown arrow. The dropdown shows the response: "Hello!! My name is Hodor. Am a Virtual Assistant of iNeuron Restaurant. How can i help you?". Another red arrow points from this section to a red circle containing a question mark icon. Below this, there's an "INTENT" section with a dropdown arrow, showing "Default Welcome Intent" with a red arrow pointing to it and a red circle containing a question mark icon. Further down, there's an "ACTION" section with the value "input.welcome", an "SENTIMENT" section with "Query Score: 0.2", and a "DIAGNOSTIC INFO" section.

1. User typed “Hello”
2. We got the response from our bot which we had added in our responses.
3. And It’s belonging to the ‘Default Welcome Intent’.

From above, we assured that our intent is working fine.

Now, we have to add some other functionalities in our restaurant chatbot like book tables, order foods, etc. For these things, we have to create more intents and will add entities as well.



Click on the given sign for creating more intents.

We created our intent for table booking.

• Book\_Table

Contexts ?

Events ?

Training phrases ?

Add user expression

PARAMETER NAME	ENTITY	RESOLVED VALUE
number	@sys.number	5
date-time	@sys.date-time	evening.

Is it possible to book table at 7:00PM?  
Can you book a table for 2 persons on Friday?

1. We named our intent as 'Book\_Table'.
2. We're adding 'Training Phrases' which will be used by User.
3. These are the default entities in Dialogflow. It shows that '2' belongs to '@sys.number' and 'evening' belongs to '@sys.date-time'

We added Responses as well in the intents.

Action and parameters

Enter action name

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	date-time	@sys.date-time	\$date-time	<input type="checkbox"/>
<input type="checkbox"/>	number	@sys.number	\$number	<input type="checkbox"/>
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

Responses

DEFAULT +

Text Response

- Sorry, We're full at 7:00PM. If any cancellation happens will inform you!
- We booked the table for you. You're most Welcome at our restaurant.
- Yes, We can book a table for 5 persons.
- Yes, Booking is available.
- We're happy to assist you. Booking is available!

6 Enter a text response variant

1. By default, entities added inside the 'Action and parameters'
2. We added our bot responses for this intent.

After, adding these things will save our intent with 'SAVE' button.

- Book\_Table



Can you book a table for 2 persons on Friday?

#### Action and parameters

Enter action name

Will check our intent is working or not.

Try it now

**Agent**

**USER SAYS** Is it possible to book table at 7:00PM?

**DEFAULT RESPONSE** We're happy to assist you. Booking is available!!

**INTENT** Book\_Table

**ACTION** Not available

PARAMETER	VALUE
number	
date-time	2021-08-18T19:00:00+0 5:30

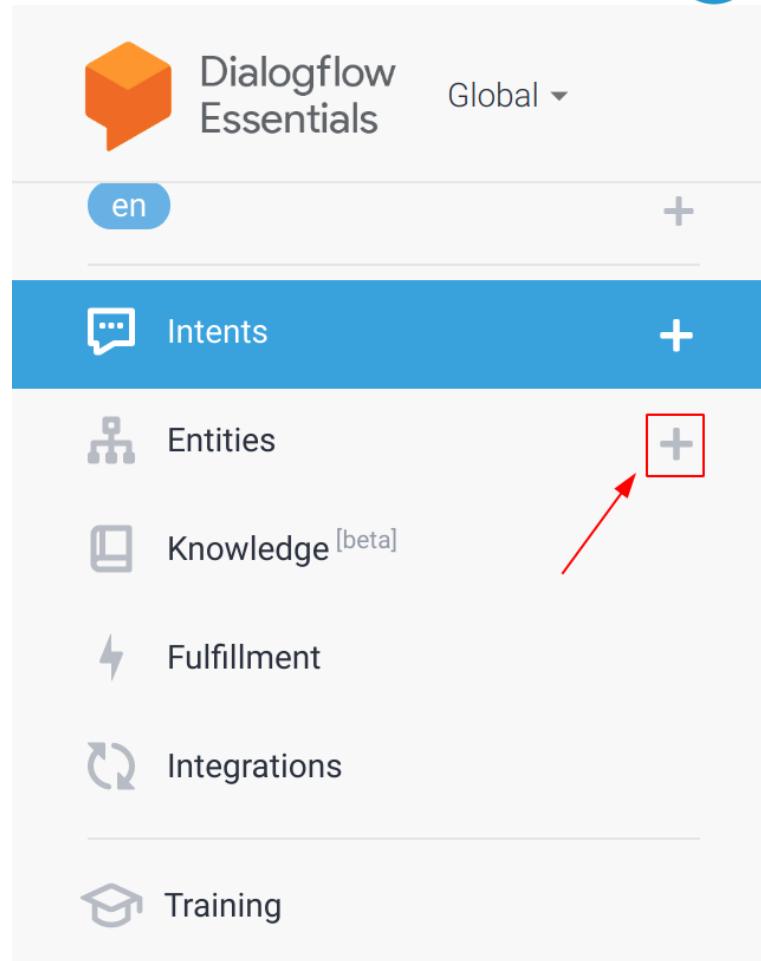
**SENTIMENT** Query Score: -0.1

**DIAGNOSTIC INFO**

So, we got that our agent is working fine for this intent.

#### Step4. Create an Entity

We have to create one more intent for ordering food but before the intent we have to add some entities also.



Click on the arrow, you'll land up into this page.

This screenshot shows the "Entities" section of the Dialogflow interface. A search bar at the top contains the text "food", which is highlighted with a red box and a red arrow pointing to it from the previous slide. Below the search bar are several checkboxes: "Define synonyms" (checked), "Regex entity" (unchecked), "Allow automated expansion" (unchecked), and "Fuzzy matching" (unchecked). A note below the checkboxes says "Separate synonyms by pressing the enter, tab or ; key." There is a "SAVE" button and a more options button "...". Below these controls is a table with five rows, each containing a text input field with the placeholder "Click here to edit entry". At the bottom of the table is a "+ Add a row" link.

'food' is my entity name here.

food

Define synonyms ?  Regexp entity ?  Allow automated expansion  Fuzzy matching ?

Biryani	Biryani
Khichdi	Khichdi
Tandoori chicken	Tandoori chicken
Paneer Tikka	Paneer Tikka
Pizza	Pizza
Manchurian	Manchurian
Momos	Momos
Chole bhature	Chole bhature
Rajma Chawal	Rajma Chawal

Click here to edit entry

SAVE

1. We added food entities in here.
2. 'SAVE' your 'food' entities.

Now, will create our new intent for ordering food.

Order\_food

Contexts ?

Events ?

Training phrases ?

Add user expression

Please, add a box of Kaju katli for me.

Is Tandoori Chicken available here?

Can you add 2 plate momos and Burfi in dessert?

Are you serving Biryani?

SAVE

We know from above which button we have to choose for creating an intent.

1. 'Order\_food' is my intent name.
2. The 'Training Phrases' added which will be Used by 'User'.
3. We can see these are the entities' item we added in there.

- Order\_food

Action and parameters

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	food	@food	\$food	<input type="checkbox"/>
<input type="checkbox"/>	number	@sys.number	\$number	<input type="checkbox"/>
<input type="checkbox"/>	Sweets	@Sweets	\$Sweets	<input type="checkbox"/>
Enter name		Enter entity	Enter value	<input type="checkbox"/>

+ New parameter

Responses

DEFAULT +

Text Response

- 1 Yes, We're serving \$food
- 2 Yes, \$Food is available. Would you like to add deserts also?
- 3 Sure, \$Sweets and \$number plate momos added. Anything more you want to add?
- 4 \$Sweets box is ready. Anything more you want?

5 Enter a text response variant

ADD RESPONSES

1. Entities which we had created added inside the 'Action and parameters'.
2. These are the entities we created, @food, @sweets , and @sys.number is the default one.
3. Responses added for our bot.
4. Lastly, 'SAVE' our intent.

Try it now

Agent

USER SAYS

Is Tandoori Chicken available here?

[COPY CURL](#)

DEFAULT RESPONSE

Yes, Tandoori chicken is available. Would you like to add deserts also?

INTENT

[Order\\_food](#)

ACTION

*Not available*

PARAMETER

VALUE

Sweets

Tandoori chicken

food

number

SENTIMENT

Query Score: 0.0

So, our intent is working fine. Now will integrate our chatbot.

### Step5. Integrating Chatbot to Telegram

1. Do Login in Telegram (If you haven't created account create it and then login).
2. Search for 'Botfather' and select the verified account of 'Botfather' with blue tick.

←  X

Chats Media Links Files Music Voice

BotFather

Chats and Contacts

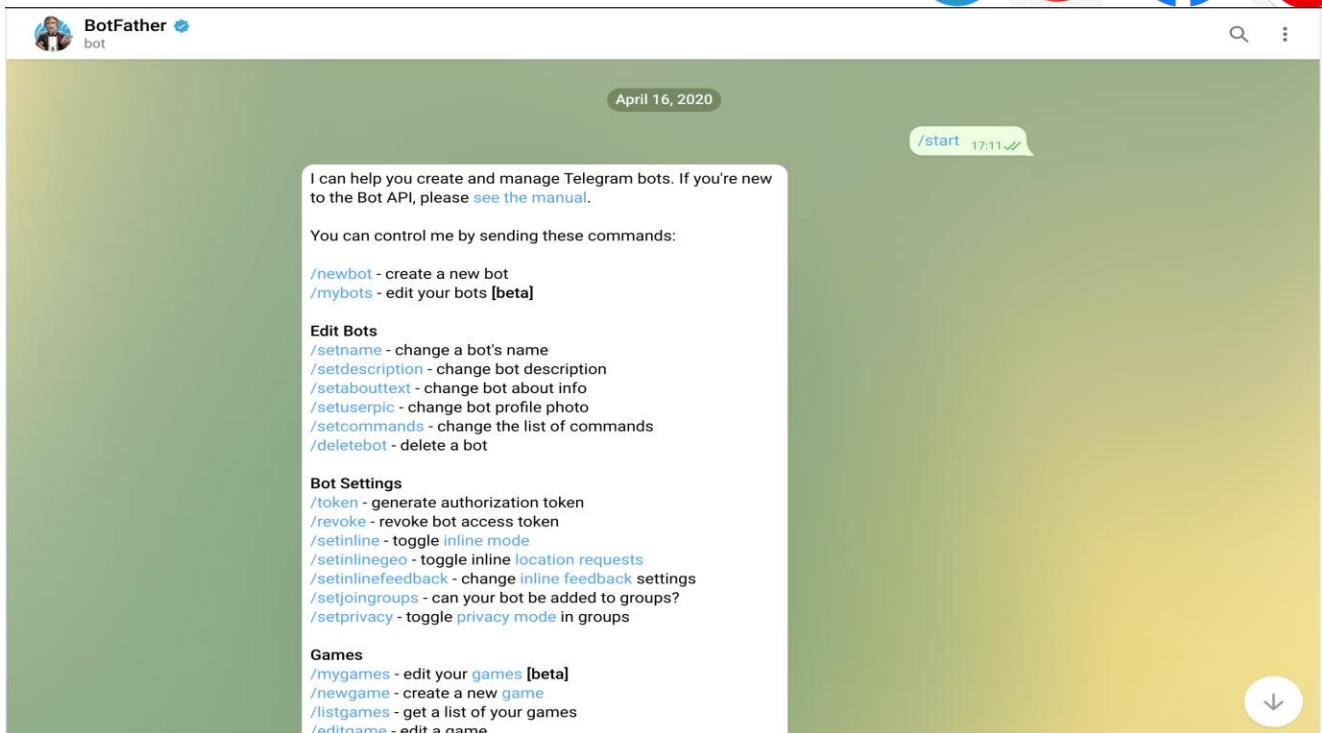
BotFather ✅  
bot

Global Search Show more

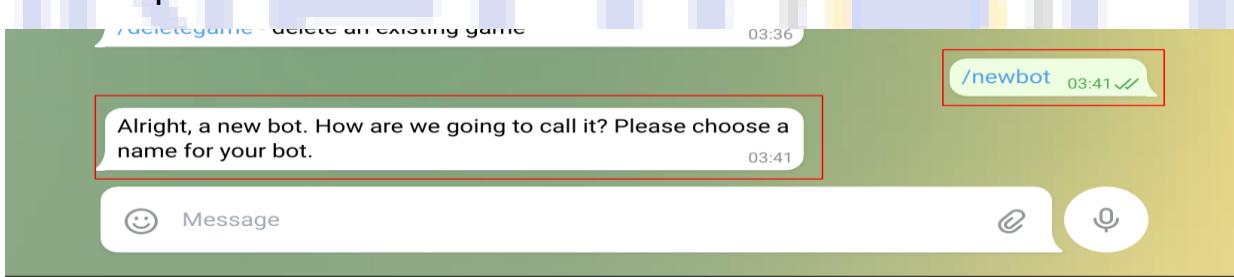
The BotFather  
@tetris101bot, bot

BotFather ✓  
@Bott\_Father, 5 264 subscribers

3. Click on 'Botfather' and type '/start'. You'll get the steps for creating a new bot in Telegram.



4. Now, for creating a new bot type '/newbot'. Will get a message in response.



5. We are adding botname as 'iNeuron\_Restaurant' and after that we have to add username also. So, we added 'Deepcafe\_bot'.

Alright, a new bot. How are we going to call it? Please choose a name for your bot.

03:41

iNeuron\_Restaurant 03:47✓

Good. Now let's choose a username for your bot. It must end in 'bot'. Like this, for example: TetrisBot or tetris\_bot.

03:47

Restaurant\_cafe\_bot 03:48✓

Sorry, this username is already taken. Please try something different.

03:48

Deepcafe\_bot 03:48✓

Done! Congratulations on your new bot. You will find it at [t.me/Deepcafe\\_bot](https://t.me/Deepcafe_bot). You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

~~Use this token to access the HTTP API:~~

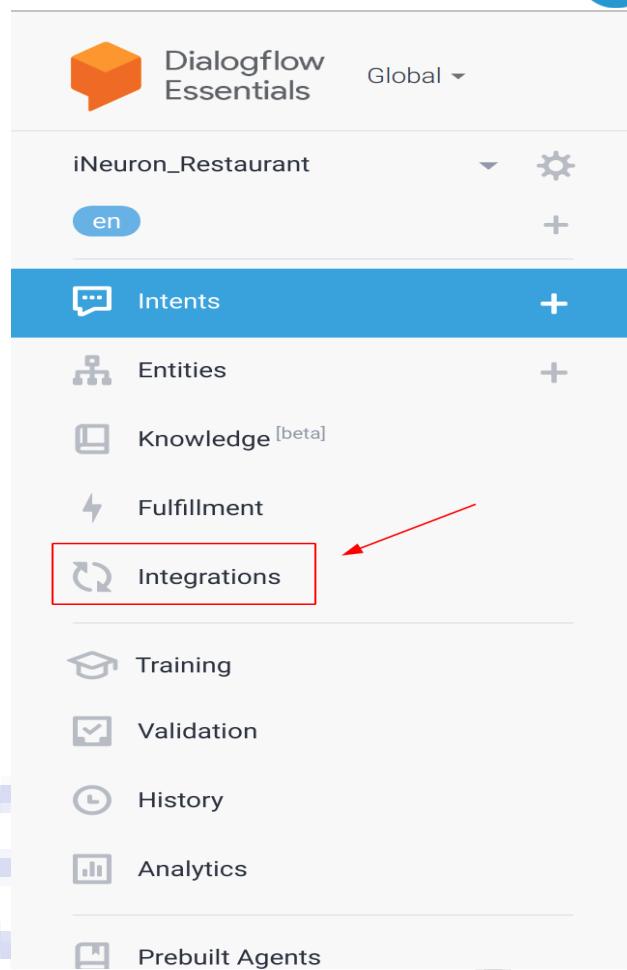
~~Keep your token secure and store it safely, it can be used by anyone to control your bot.~~

For a description of the Bot API, see this page:

<https://core.telegram.org/bots/api>

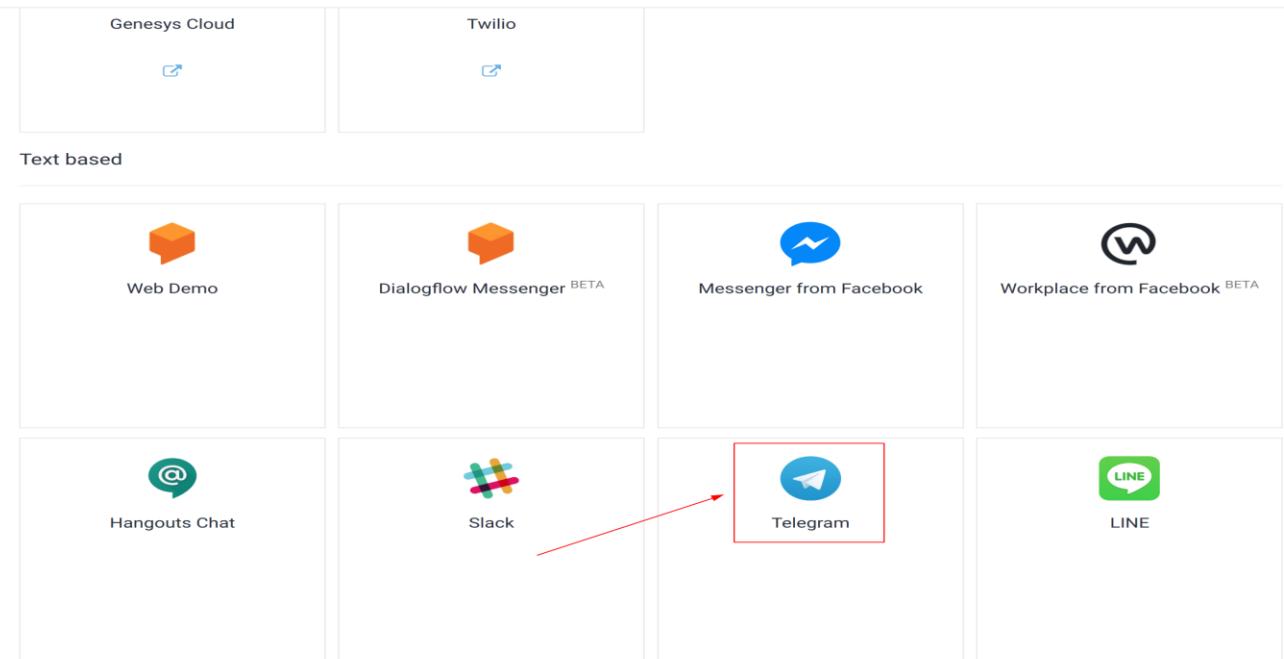
03:48

- a. After, finalization of your bot name we will get one token that we have to add in Dialogflow Telegram. From there our bot will be active in telegram.
6. Now, will integrate in Dialogflow. We have to click on 'Integration'.

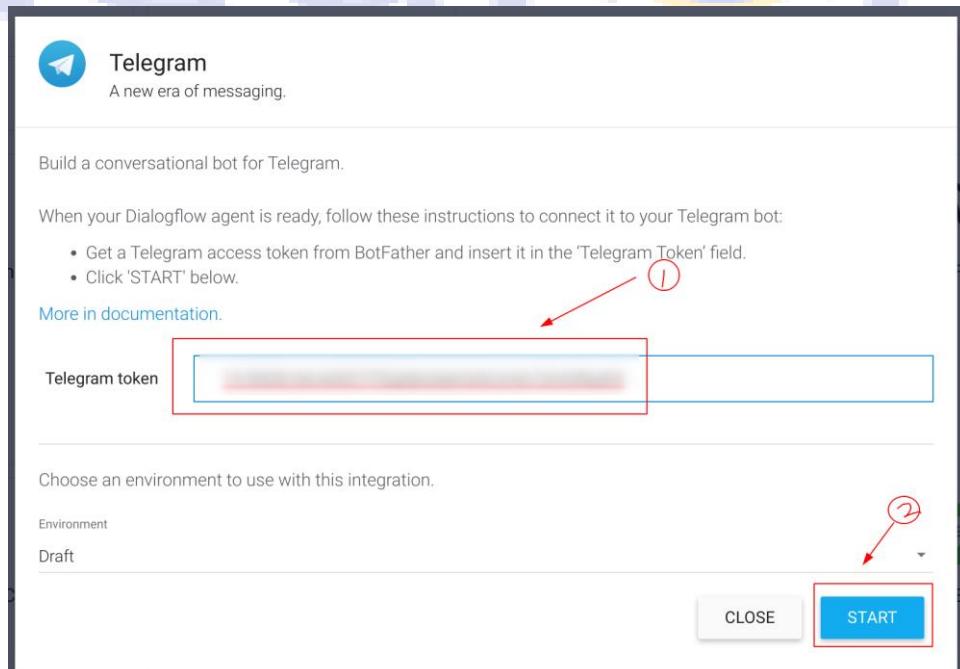


7. We have to find out 'Telegram' which is under 'Text based'.

## Integrations



8. Select 'Telegram', then add your Telegram token inside here and press on 'Start'.



9. Now, go to your telegram and click on the given arrow or you can find your bot in search.

Done! Congratulations on your new bot. You will find it at [t.me/Deepcafe\\_bot](https://t.me/Deepcafe_bot). You can now add a description, about section and profile picture for your bot, see /help for a list of commands. By the way, when you've finished creating your cool bot, ping our Bot Support if you want a better username for it. Just make sure the bot is fully operational before you do this.

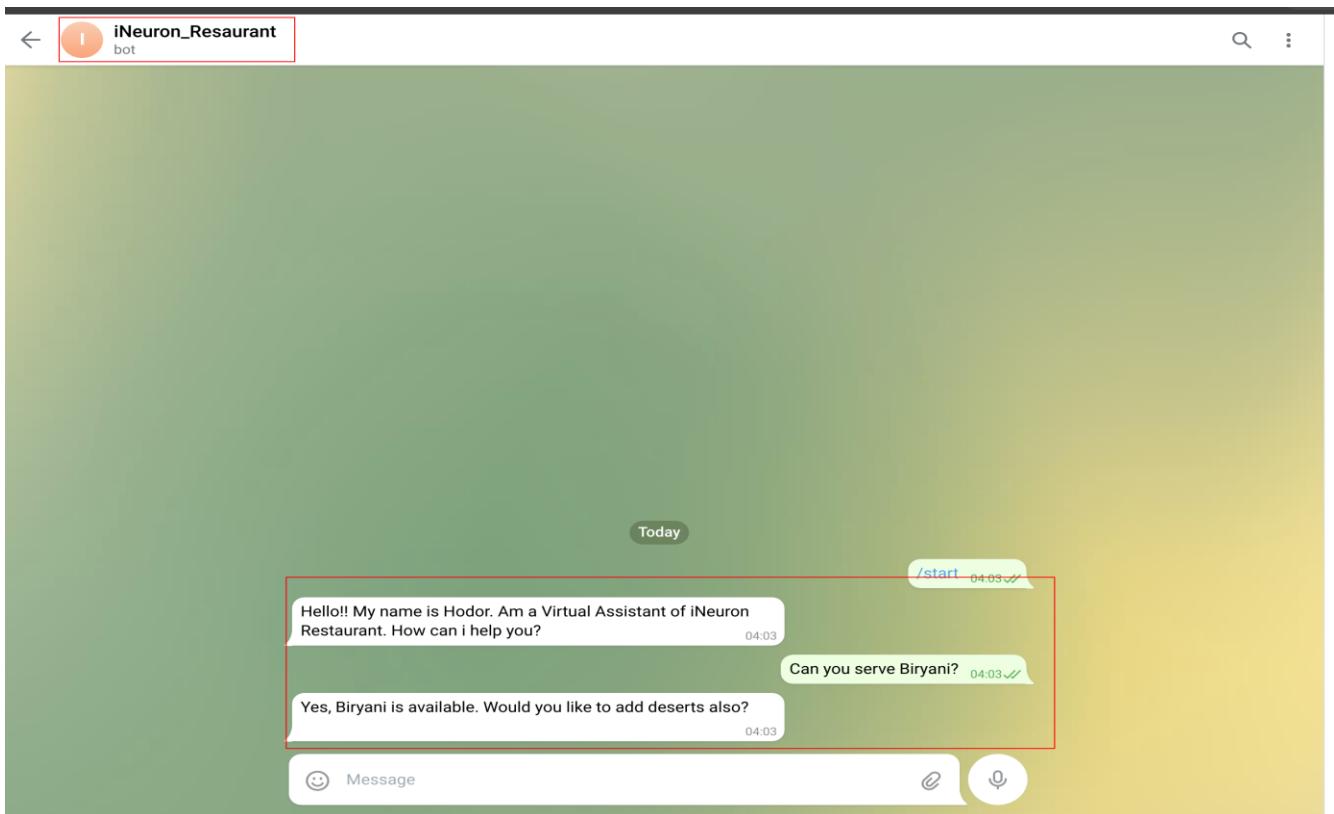
Use this token to access the HTTP API:

Keep your token **secure** and **store it safely**, it can be used by anyone to control your bot.

For a description of the Bot API, see this page:

<https://core.telegram.org/bots/api>

03:48

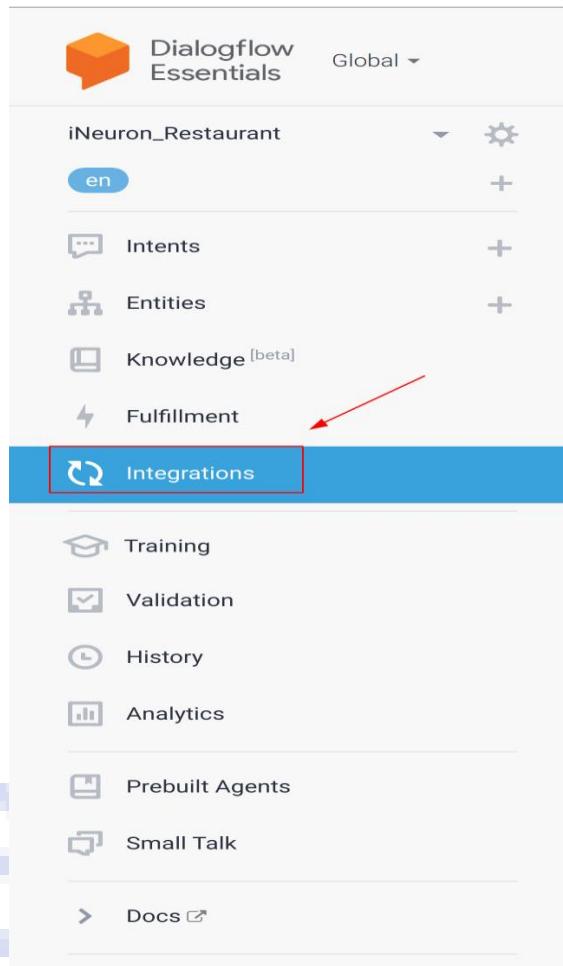


Now, we got that our bot is working fine in Telegram. You can get our bot at here [iNeuron Restaurant](#)

## Integrate Chatbot to Facebook Messenger

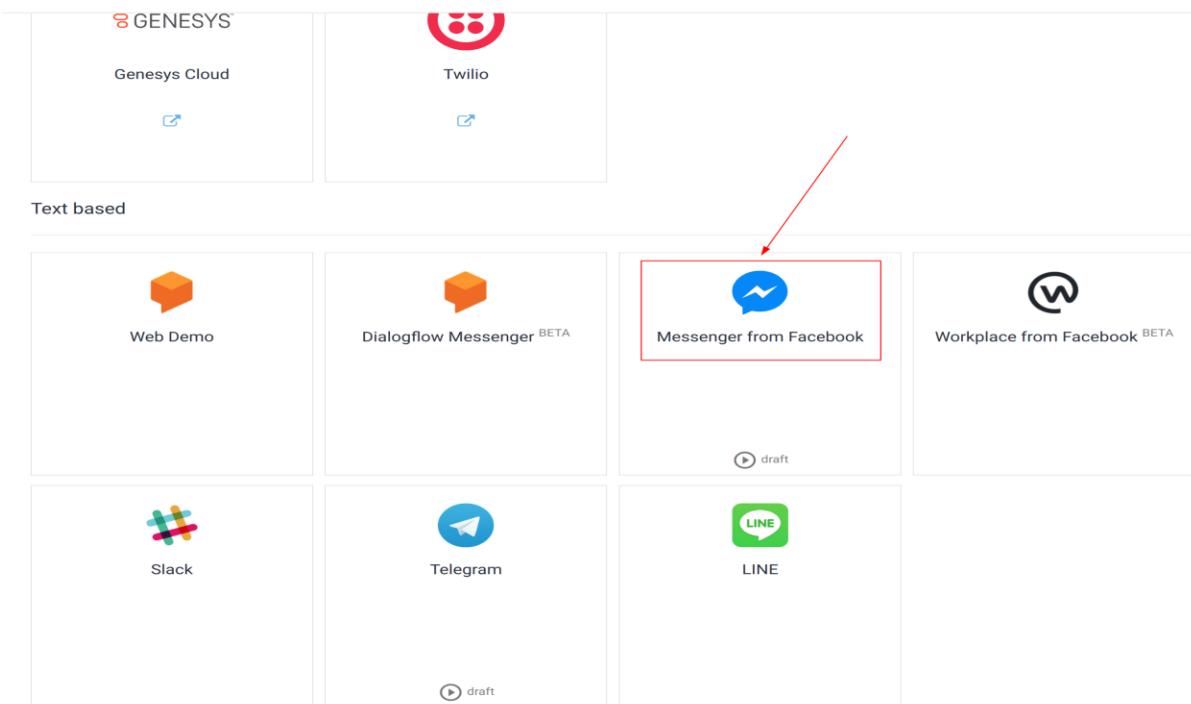
We can integrate Dialogflow to Facebook for promoting our Chatbot to different Platform.

**Step1.** Do Setup the integration



Click on the arrow, you'll land into the below page then find 'Messenger from Facebook' in there and select it.

## Integrations



### Step2. After clicking on 'Messenger from Facebook', will get a pop-up.

**Facebook Messenger**  
An easier way to message.

Create and teach a conversational bot for Facebook Messenger.

After you design and test your Dialogflow agent, you can launch your Messenger bot

1. Get your Facebook Page Access Token and insert it in the field below.
2. Create your own Verify Token (can be any string).
3. Click 'START' below.
4. Use the Callback URL and Verify Token to create an event in the Facebook Messenger Webhook Setup.

[More in documentation.](#)

Show old callback URL (more details [here](#))

Callback URL:

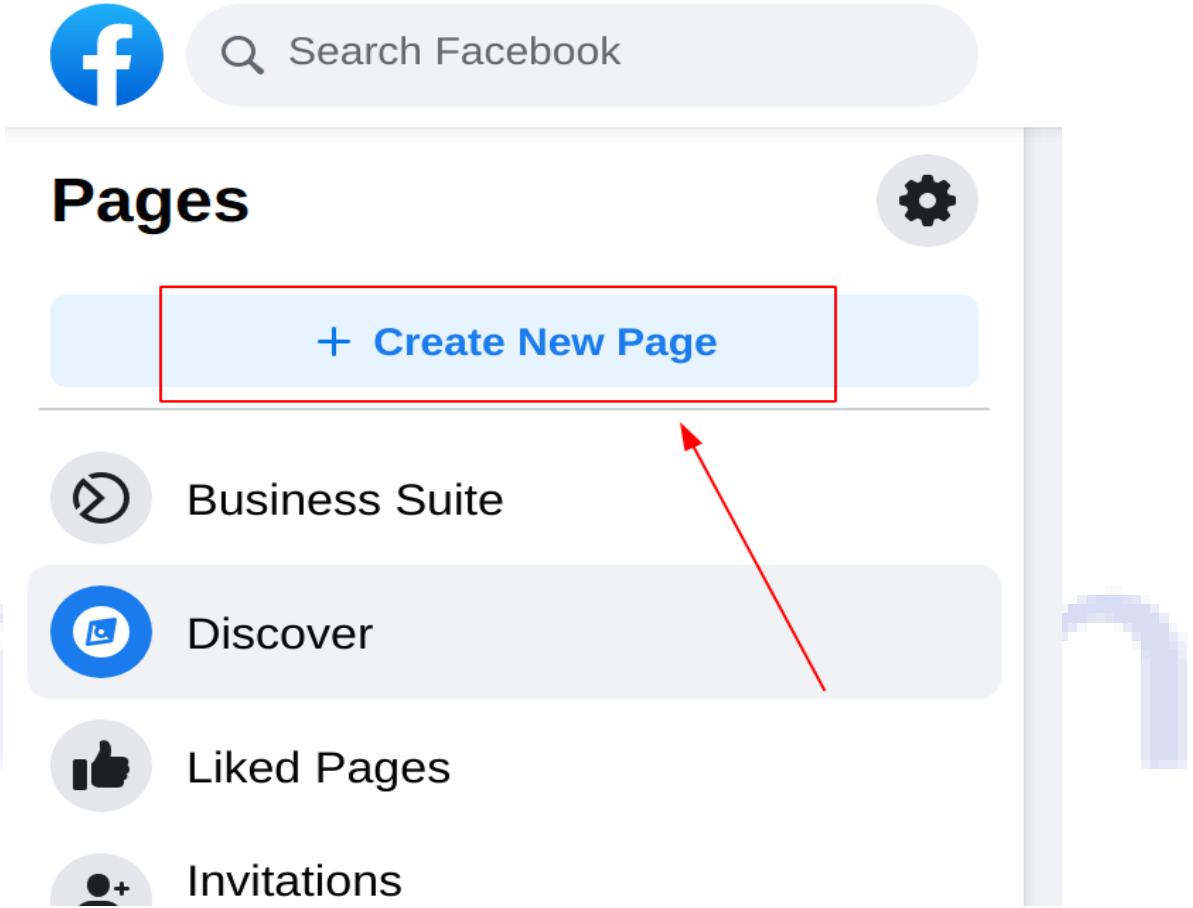
Verify Token:

Page Access Token:

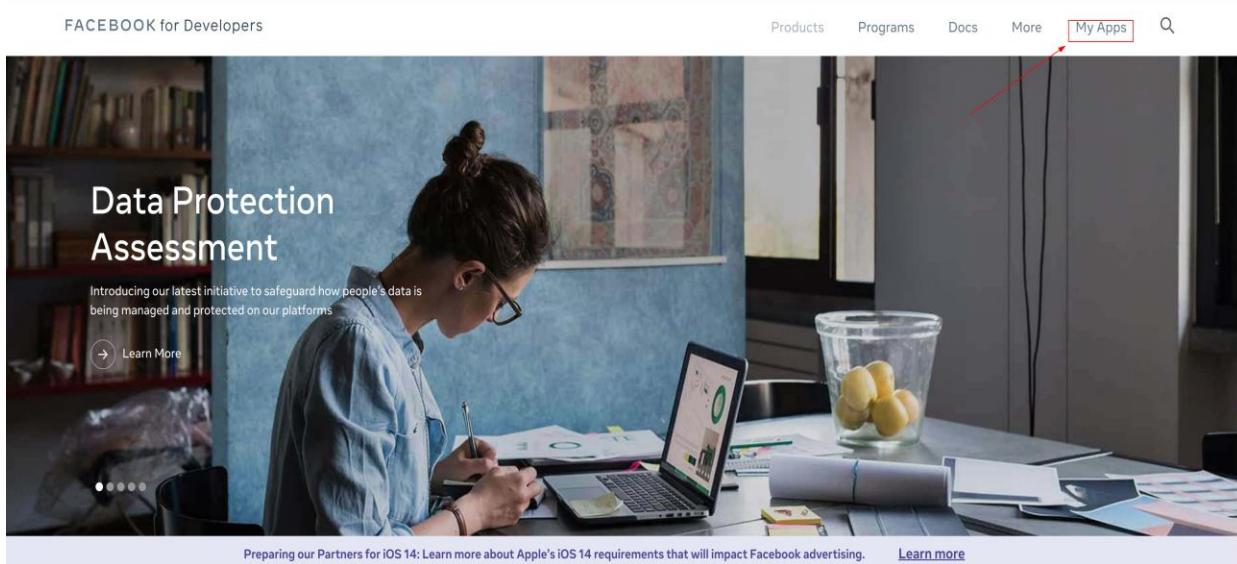
Active environment:

Click here to enter text.

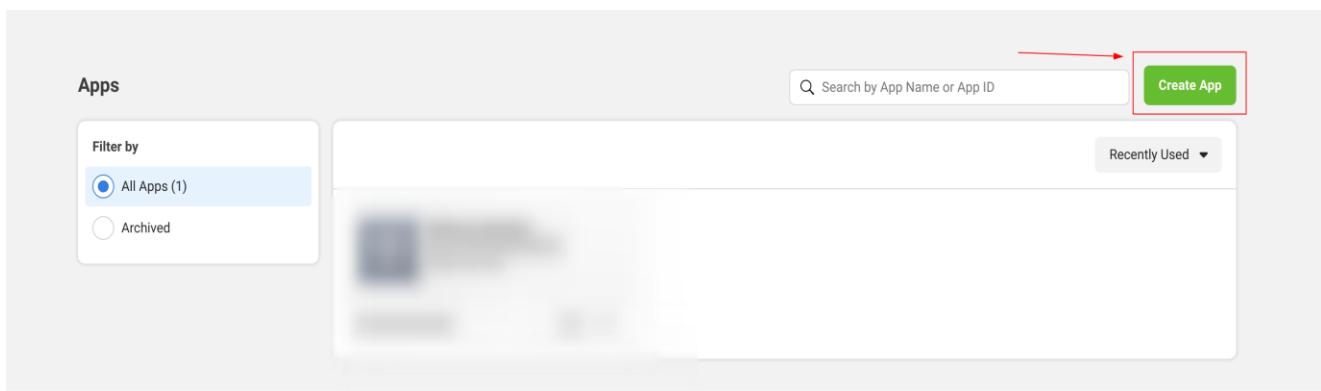
**Step3.** First do signin in Facebook, and create a new Page.



**Step4.** To create a bot, we have to register ourselves in Facebook Developer page through this [link](#)



Click on 'My Apps' will be forwarded to below image.



Then we have to click on 'Create App' and Select 'Business' and fill up the required filled.

**Create an App**

**App Display Name** (1)  
This is the app name associated with your app ID.

**App Contact Email** (2)  
This email address is used to contact you about potential policy violations, app restrictions or steps to recover the app if it's been deleted or compromised.

**App Purpose** (3)  
This app's primary purpose is to access and use data from Facebook's Platform on behalf of:  
 Yourself or your own business  
 Clients (i)

If you are developing an app that accesses and uses data from Facebook's Platform on behalf of clients, you are subject to [Section 5b of the Platform Terms](#).

**Do you have a Business Manager account? · Optional**  
In order to access certain aspects of the Facebook platform, apps may need to be connected to a verified Business Manager account. If you haven't yet set up an account, you can create one now or later in the process.

(4)

By proceeding, you agree to the [Facebook Platform Terms](#) and [Developer Policies](#).

[Back](#) **Create App**

1. You can add your App name in here as we taken 'Test'.
2. Provide your mail-id.
3. App Purpose select 'Yourself'.
4. Last click on, 'Create App'
5. Then, it'll ask for your Facebook Password.

After creating app will land into this page.

FACEBOOK for Developers

App ID: [REDACTED] App Type: Business

across apps, devices, platforms and websites.

Facebook advertisers.

The world's number one social login product.

**Test**

Dashboard

Settings

Roles

Alerts

App Review

Products Add Product

Activity Log

Fundraisers

Instagram Graph API

Jobs

Marketing API

Messenger

Web Payments

Read Docs Set Up

We can see, our 'Test' bot is reflecting in there with 'App ID'.

1. Select the 'Messenger' and click on 'setup'.

Then, Will land in Permission Page.

To get started, please read the [developer documentation](#) for complete details. Your access to and use of the Platform are subject to [Facebook Platform Terms](#) and [Developer Policies](#), together with all other applicable terms and policies.

**Get Started**

Check out some resources that may help you in development: [Quick Start](#) (build a Messenger app in 10 minutes) and [Complete Documentation](#). Also join our [Messenger Platform Developers Community](#) on Facebook to get latest news and learn more!

**Access Tokens**

Generate a Page access token to start using the platform APIs. You will be able to generate an access token for a Page if:

1. You are one of the Page admins, and
2. The app has been granted the Page's permission to manage and access Page conversations in Messenger.

Note: If your app is in dev mode, you can still generate a token but will only be able to access people who manage the app or Page.

**No page permissions granted**

You'll need to connect pages and grant them the required permissions in order for tokens to be generated.

[Add or Remove Pages](#)

1. If you didn't create your page create it from here.

2. If Page is available, just the 'Add or Remove Pages' and select your page.

**Access Tokens**

Create New Page

Generate a Page access token to start using the platform APIs. You will be able to generate an access token for a Page if:

1. You are one of the Page admins, and
2. The app has been granted the Page's permission to manage and access Page conversations in Messenger.

Note: If your app is in dev mode, you can still generate a token but will only be able to access people who manage the app or Page.

Pages ↑	Tokens
 Test 106578465077061	—

Add or Remove Pages ⓘ

**Generate Token**

1. After selecting your Page, it'll look like above as I created page with name as 'Test'.
2. It will 'Generate Token' which will be added in Dialogflow.

**Token Generated**

 Test  
106578465077061

To protect your security, ONLY share this token with app developers you trust.

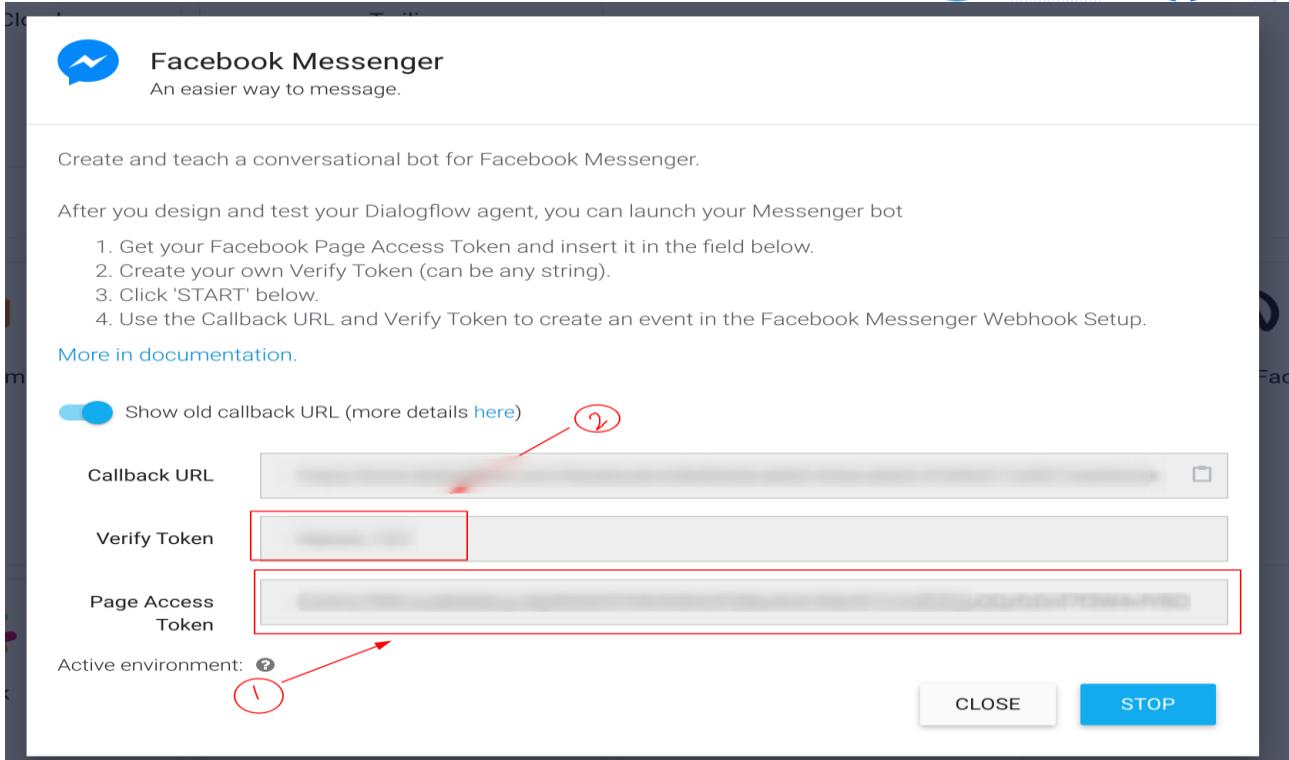
This token will only be shown once, so keep it safe. If it gets lost, you'll need to create a new one. Anyone could potentially use this token to impersonate this page, depending on the privacy settings of your app. If you wish to revoke all previously generated tokens for this app, you can go to Business Integrations settings. [Learn More](#)

I Understand

Copy

Done

Click on 'Generate Token' will get the above pop-up. Copy your 'Token' and paste in Dialogflow as below.



1. Place your Facebook token in 'Page Access Token' in Dialogflow 'Messenger from Facebook'
2. Add verify token any string like we had taken 'Daisy\_123'. And click on 'START'.

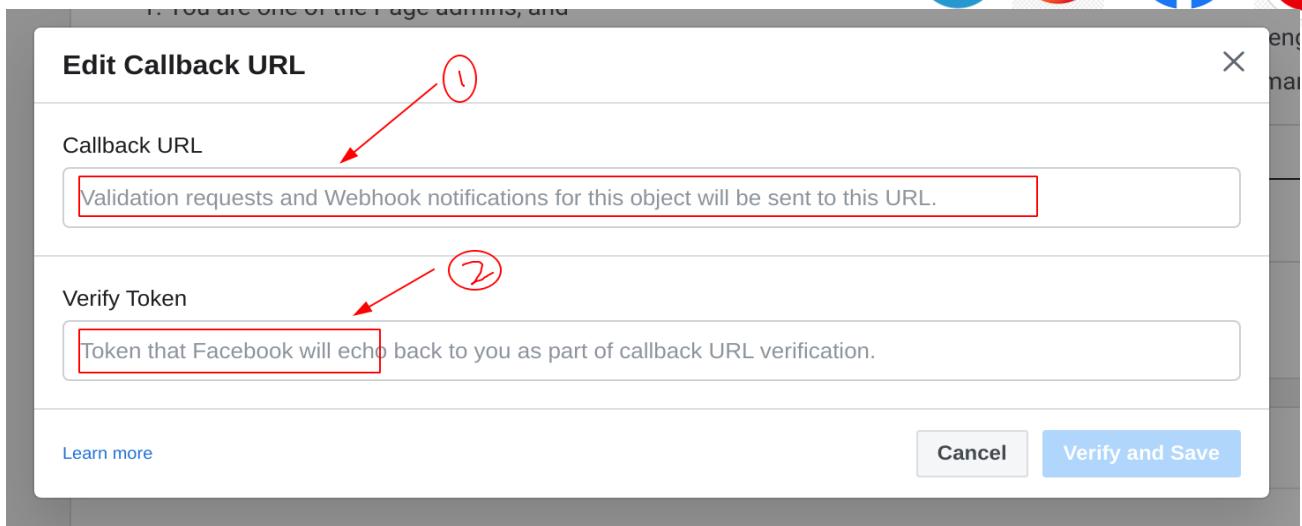
Then, we have to add a Callback url from Dialogflow and added in Facebook Webhook.

The screenshot shows the Facebook for Developers dashboard for a 'Business' app. On the left, there's a sidebar with options like Dashboard, Settings, Roles, Alerts, App Review, Products, Add Product, Messenger (selected), Settings, Instagram Settings, Activity Log, and Activity Log. The main area has two tabs: 'Access Tokens' and 'Create New Page'. Under 'Access Tokens', it says 'Generate a Page access token to start using the platform APIs. You will be able to generate an access token for a Page if: 1. You are one of the Page admins, and 2. The app has been granted the Page's permission to manage and access Page conversations in Messenger.' It lists a page named 'Test' with ID '106578465077061' and a 'Generate Token' button. Below that is a 'Webhooks' section with a note: 'To receive messages and other events sent by Messenger users, the app should enable webhooks integration.' A red box highlights the 'Add Callback URL' button.

**Copy the 'Callback URL' from Dialogflow 'Messenger from Facebook' and add that in Facebook Webhook.**

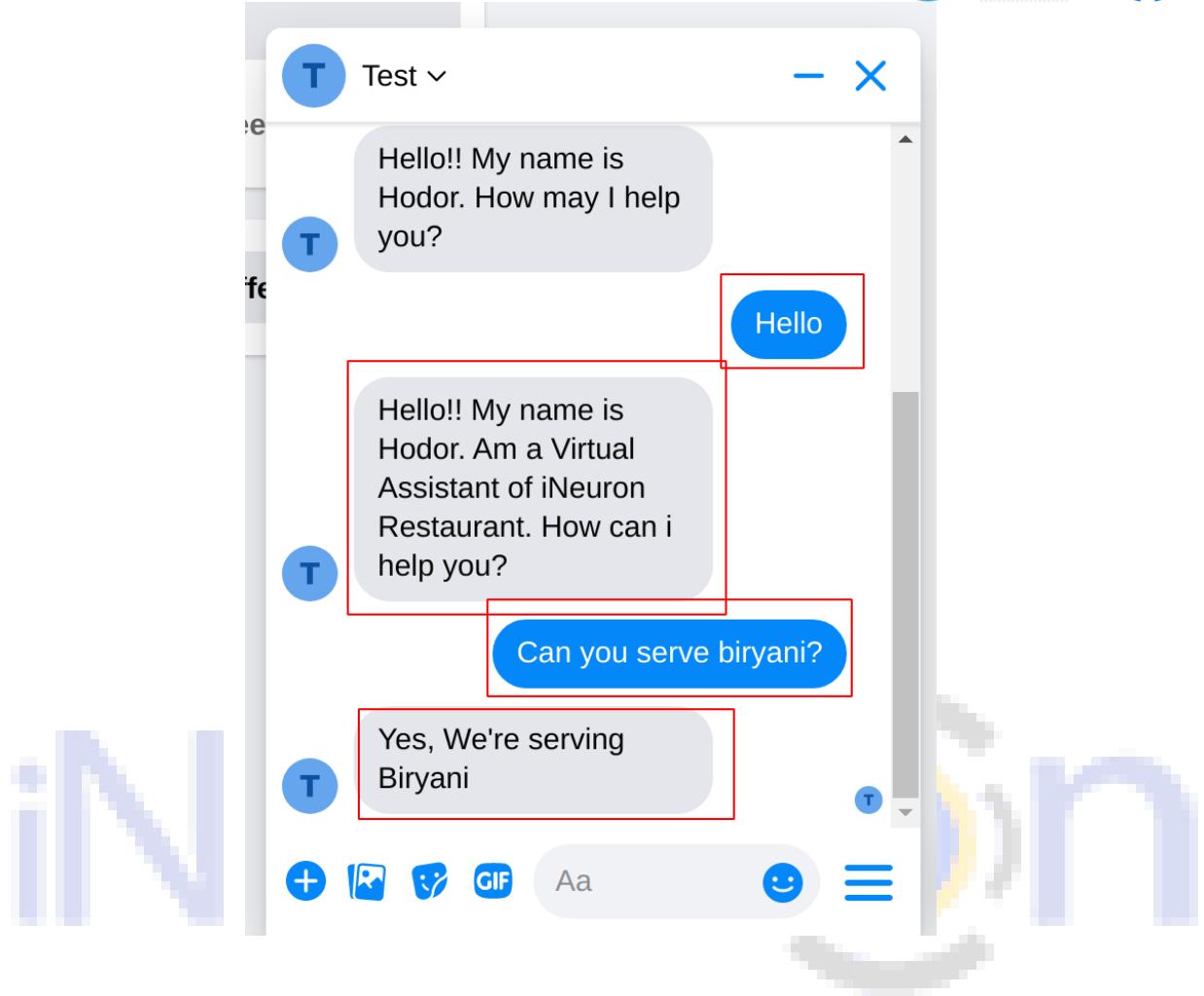
The screenshot shows a Dialogflow agent setup dialog for 'Facebook Messenger'. It includes instructions: 'Create and teach a conversational bot for Facebook Messenger.', 'After you design and test your Dialogflow agent, you can launch your Messenger bot.', and a list of steps: '1. Get your Facebook Page Access Token and insert it in the field below.', '2. Create your own Verify Token (can be any string).', '3. Click 'START' below.', and '4. Use the Callback URL and Verify Token to create an event in the Facebook Messenger Webhook Setup.' A red arrow points to the 'Callback URL' input field, which contains a blurred URL. Other fields include 'Verify Token' (containing 'abcde...') and 'Page Access Token' (containing a long token). Buttons at the bottom are 'CLOSE' and 'STOP'.

**Add the 'Callback URL' in Webhook.**



1. Paste the 'Callback URL' in here.
2. In Dialogflow we had taken 'Daisy\_123' same name we have to add in here.
3. Now, click on 'verify and save' and Our bot is ready.

Let's check in our Facebook Page our bot is giving us a response or not.

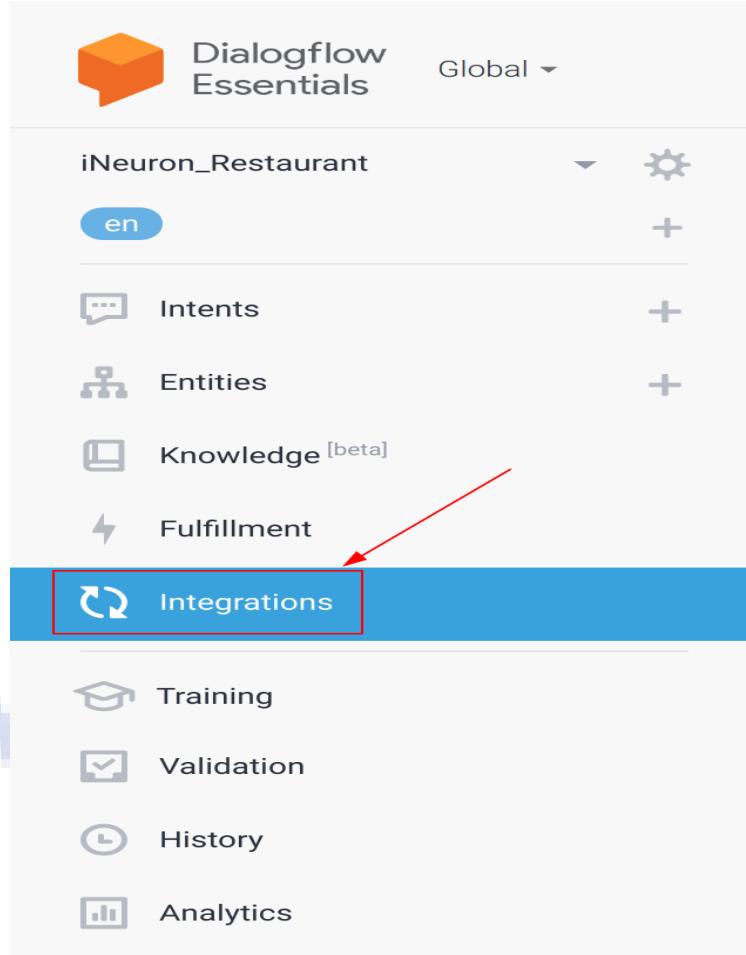


We can check above. Our bot is working fine and giving us a response.

### Integrate Chatbot with 'Slack'

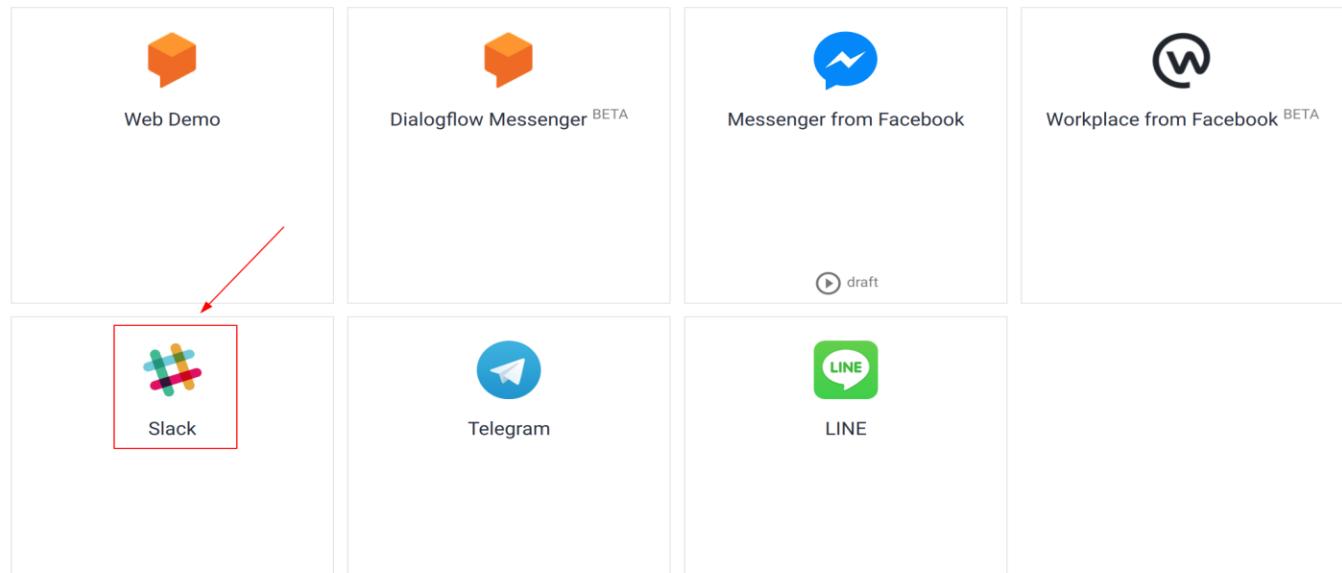
For a business, we have to promote our restaurant\_bot to every platform. We're integrating our bot with 'Slack'.

**Step1.** In the Dialogflow, check on the left side ‘Integrations’.



**Step2.** Look for ‘Slack’ and select that.

Text based



After clicking on 'Slack' window will pop-up like



## Slack

Build a conversational bot for Slack.

### Test

To test your integrated agent, connect it to your Slack account by clicking 'Test in Slack' and then signing in with your team account.

[TEST IN SLACK](#)

---

### Launch

To make your bot public, take the following steps:

- Create a new [Slack app](#).
- In the Slack app, go to the 'Bot users' section and click the 'Add a Bot User' button.
- In the Slack app 'Basic Information' section, copy the 'Client ID', 'Client Secret' and 'Verification Token' and paste their values into the respective fields below.
- Start the bot.
- Copy the 'OAuth URL' value below and paste it into the 'Redirect URL(s)' field from the 'OAuth & Permissions' section of your Slack app settings. Save URLs.
- Copy Events Request URL from the field below and paste it to Request URL in the 'Events Subscription' section of your Slack app.
- Enable events in the 'Event Subscriptions' section.
- Select necessary events in "Subscribe to Bot Events" section for example message.im, message.group, message.channel, im\_created. These events will define which message types (direct, in channel, etc.) your bot will react to. Save changes.
- In the Slack app settings, go to 'Manage Distribution' and add the bot to your Slack team using the 'Add to Slack' button.
- To use [rich messages](#), copy the 'Events Request URL' value below and paste it into the 'Request URL' field from the Interactive Messages section of your Slack app settings.

More in the [documentation](#).

Slack Client ID \*

Slack Client Secret \*

[CLOSE](#) [CREATE](#)

## Step3. Configuring Slack

For Set-up the Slack integration for your Dialogflow agent. Two things required here:

- A [Slack Account](#)
- A Slack Team

## Step4. Create a Slack application

First, we go to the [Slack Developer Console](#).

The screenshot shows the Slack API homepage with a dark background. At the top left is the 'slack api' logo. To its right is a search bar with the placeholder 'Search'. On the far right are links for 'Docs', 'Tutorials', and 'Your apps'. Below the header, the main title 'Customize your workspace' is displayed in large white font. A subtitle below it reads 'Connect, simplify, and automate your work with Slack apps.' Three buttons are visible: 'Create an app' (highlighted with a red circle), 'Explore workflows', and 'Browse App Directory'.

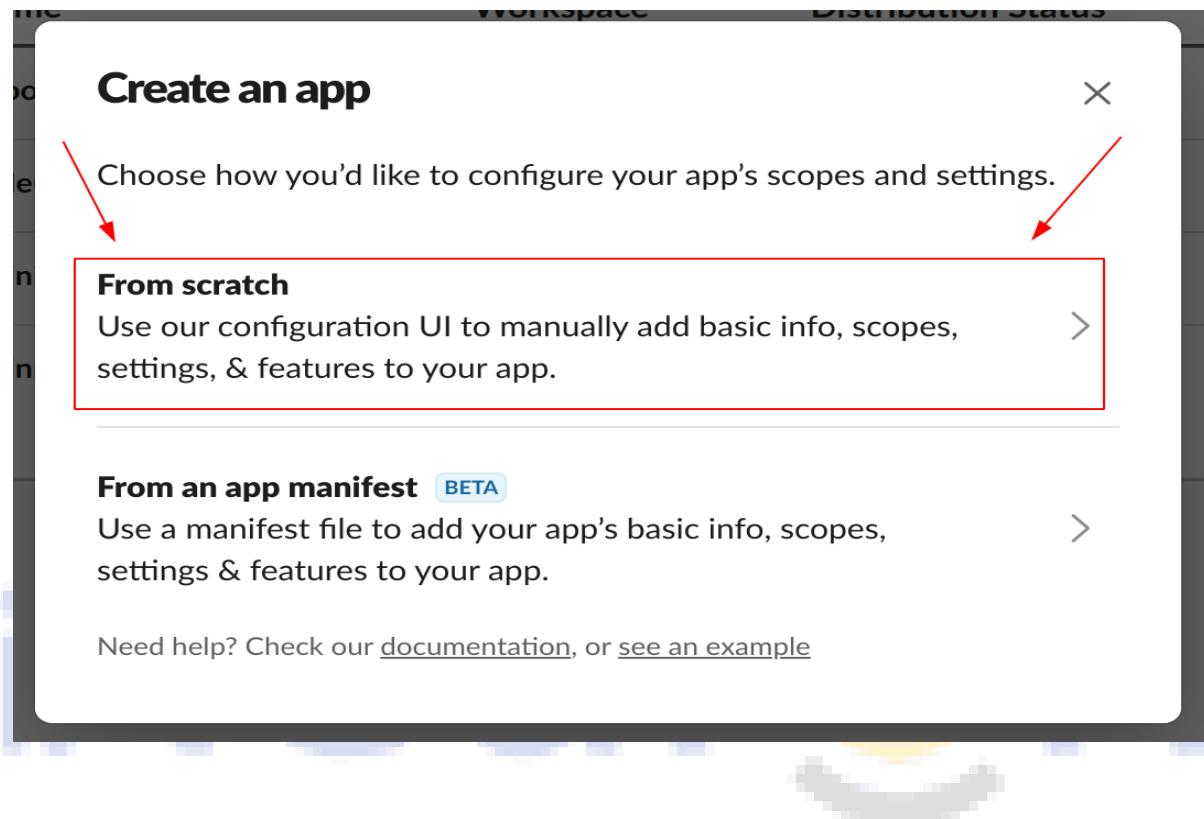
This section highlights how users can save time by automating repetitive tasks. It features a testimonial from 'Nine Publishing' about using custom tools and apps to simplify newsroom processes. Below the testimonial, a callout box states 'Reduced day-long processes to 30 minutes'. To the right, there's a preview of a workflow titled 'Save Customer Story' which starts from a 'Save customer story' shortcut in '#team-sales' and creates a form for a customer story, then adds a spreadsheet row.

The screenshot shows the 'Your Apps' section of the Slack API. On the left is a sidebar with various API categories like Start learning, Authentication, Surfaces, Block Kit, Interactivity, Messaging, APIs, Workflows, Enterprise, Apps for Admins, and Reference. The main area is titled 'Your Apps' and contains a table with columns for 'App Name', 'Workspace', and 'Distribution Status'. A green button labeled 'Create New App' is located at the top right of this section. A red box highlights the table area, and a red arrow points to the 'Create New App' button. Below the table, a message says 'Don't see an app you're looking for? Sign in to another workspace.' At the bottom, there are links for Changelog, Beta features, Developer blog, Roadmap, @SlackAPI, Resources, Our tools, Get support, Partner with us, Japanese language link, API Terms of Service, and Slack Fund.

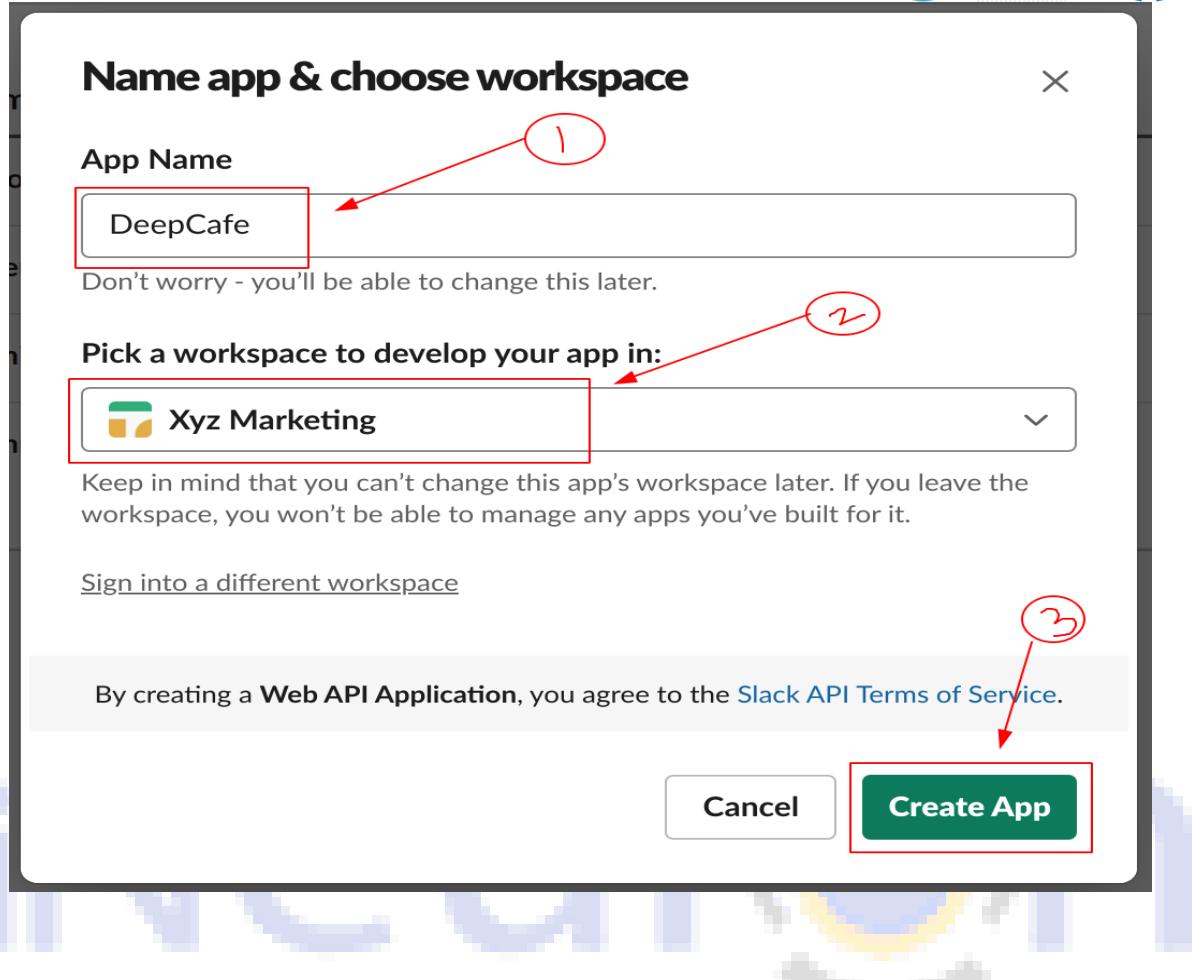
1. Click on 'Create New App' for filling some details in it.
2. Few apps are created already in here.

Click here to enter text.

After, clicking on 'Create New App' will get one pop-up select 'From scratch'



Then we have to fill some details.



1. Add your App Name in here as we had taken 'DeepCafe'.
2. Here, we have to select the workspace through the drop-down menu as we selected 'Xyz Marketing'.
3. Click on 'Create App'.

After clicking on 'Create App' will land up in the below page.

The screenshot shows the Slack API Basic Information page for an app named 'DeepCafe'. The left sidebar has a red arrow pointing to the 'Basic Information' tab under 'Settings'. The main content area is titled 'Building Apps for Slack' and includes sections for 'Add features and functionality' (with options like Incoming Webhooks, Interactive Components, Slash Commands, Event Subscriptions, Bots, and Permissions), 'Building an internal app locally or behind a firewall?', and 'Where's Bot User?'. At the bottom are 'Discard Changes' and 'Save Changes' buttons.

We can check on left hand side our App Name is reflecting in here.

## Step5. Add a Bot User

On the left-hand side '**OAuth & Permissions**' is present click on that.

Go to the 'Scopes' section

The screenshot shows the Slack App Builder interface. At the top, there are four social media icons: LinkedIn, Instagram, Facebook, and YouTube.

**OAuth & Permissions** (highlighted with a red box and arrow):

- Event Subscriptions
- User ID Translation
- Beta Features
- Where's Bot User
- ?

**Submit to App Directory**

- Review & Submit

**Slack** (highlighted with a red box and arrow):
 

- Help
- Contact
- Policies
- Our Blog

**Redirect URLs**: You will need to configure redirect URLs in order to automatically generate the Add to Slack button or to distribute your app. If you pass a URL in an OAuth request, it must (partially) match one of the URLs you enter here. [Learn more](#).

**Redirect URLs** (highlighted with a red box and arrow):
 

- You haven't added any Redirect URLs
- Add New Redirect URL
- Save URLs

**Scopes** (highlighted with a red box and arrow):
 

- A Slack app's capabilities and permissions are governed by the **scopes** it requests.
- Bot Token Scopes** (highlighted with a red box and arrow):
  - Scopes that govern what your app can access.
- OAuth Scope      Description
- You haven't added any OAuth Scopes for your Bot token.
- Add an OAuth Scope (highlighted with a red box and arrow)

Click here to enter text.



## Scopes

A Slack app's capabilities and permissions are governed by the [scopes](#) it requests.

### Bot Token Scopes

Scopes that govern what your app can access.

OAuth Scope	Description	
<a href="#">app_mentions:read</a>	View messages that directly mention DeepCafe in conversations that the app is in	
<a href="#">calls:read</a>	View information about ongoing and past calls	
<a href="#">calls:write</a>	Start and manage calls in a workspace	
<a href="#">channels:history</a>	View messages and other content in public channels that DeepCafe has been added to	
<a href="#">channels:read</a>	View basic information about public channels in a workspace	
<a href="#">chat:write</a>	Send messages as DeepCafe	
<a href="#">groups:history</a>	View messages and other content in private channels that DeepCafe has been added to	
<a href="#">im:history</a>	View messages and other content in direct messages that DeepCafe has been added to	
<a href="#">im:read</a>	View basic information about direct messages that DeepCafe has been added to	
<a href="#">im:write</a>	Start direct messages with people	
<a href="#">team:read</a>	View the name, email domain, and icon for workspaces DeepCafe is connected to	

Add an OAuth Scope

1. Select these 'OAuth Scope' from 'Add an OAuth Scope'.
2. From here we have to select above scopes for our bot.

Then, click on 'App Home' on the left-hand side menu.

DeepCafe

## App Home

**Settings**

- Basic Information
- Collaborators
- Socket Mode
- Install App
- App Manifest BETA
- Manage Distribution

**Features**

- App Home** (1)
- Org Level Apps
- Incoming Webhooks
- Interactivity & Shortcuts
- Slash Commands
- Workflow Steps
- OAuth & Permissions
- Event Subscriptions
- User ID Translation
- Beta Features
- Where's Bot User ?

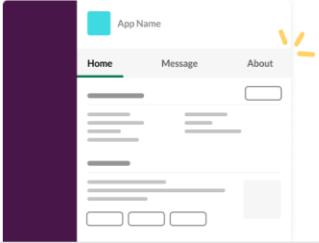
**Submit to App Directory**

Review & Submit

**Where people find your app on Slack**

Your app's home includes three tabs (Home, Messages and About) that help people have richer interactions with your app and better understand its functionality.

[Learn more](#)



**Your App's Presence in Slack**

Your app display name and presence is how your users will see your app in Slack.

**App Display Name**

**Edit** (2)

Display Name (Bot Name): DeepCafe  
Default Name: deepcafe

**Always Show My Bot as Online**

**Bot Token Added**



1. Click on 'App Home'.
2. We can edit our App Name, if we want.
3. Change your bot to always online.

## Step6. Using Dialogflow to Enable Integration

In the Slack Developer Console, click on the left-hand side select 'Basic information' and scroll down to the 'App Credentials' Section.

## App Credentials

These credentials allow your app to access the Slack API. They are secret. Please don't share your app credentials with anyone, include them in public code repositories, or store them in insecure ways.

**App ID**

**Date of App Creation**

August 23, 2021

**Client ID**

**Client Secret**

 Show Regenerate

You'll need to send this secret along with your client ID when making your [oauth.v2.access](#) request.

**Signing Secret**

 Show Regenerate

Slack signs the requests we send you using this secret. Confirm that each request comes from Slack by verifying its unique signature.

**Verification Token**

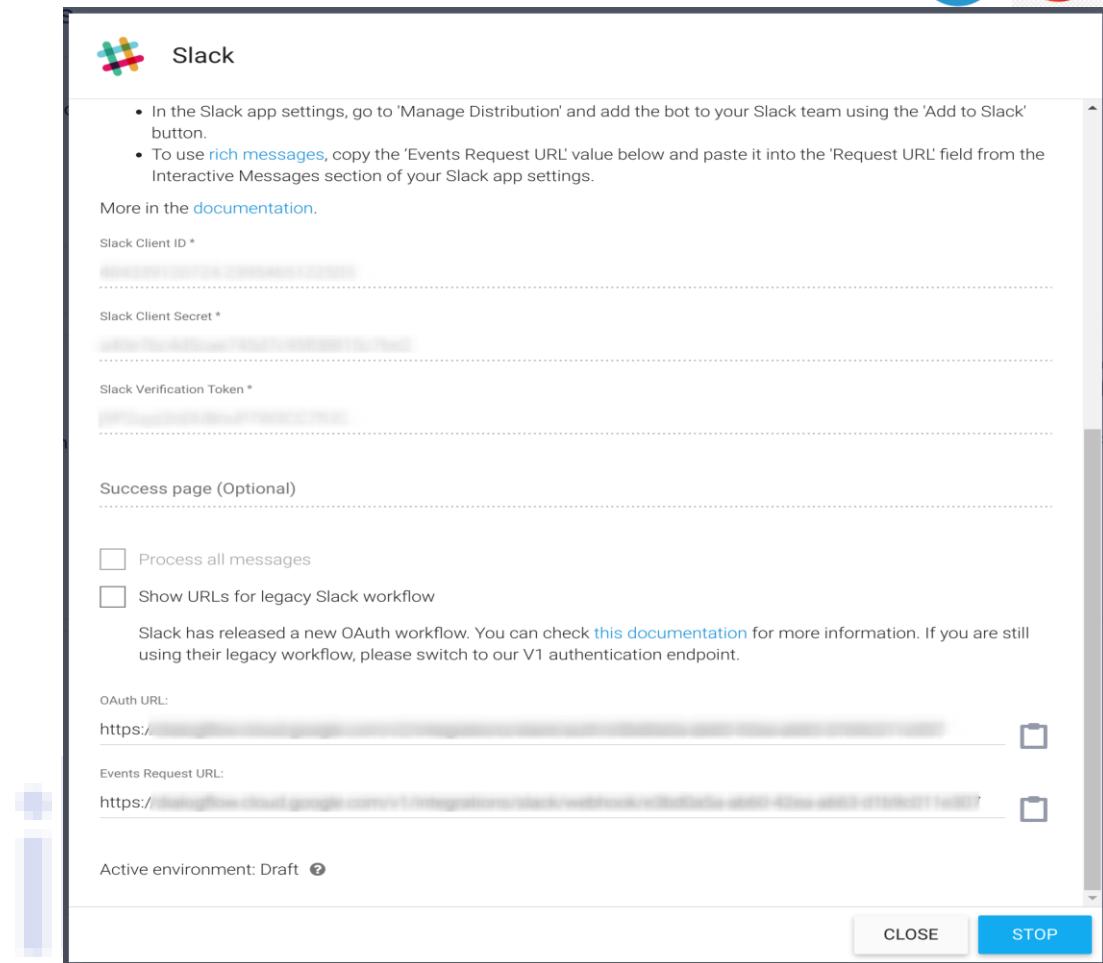
 Regenerate

This deprecated Verification Token can still be used to verify that requests come from Slack, but we strongly recommend using the above, more secure, signing secret instead.

‘Client ID’, ‘Client Secret’ and ‘Verification Token’ are required for our Dialogflow agent.

### Step7. Link with Dialogflow

In Dialogflow, go the ‘Integration’ and select ‘Slack’ tile in there.



We need to add ‘Client ID’, ‘Client Secret’ and ‘Verification Token’ inside the Dialogflow from Slack Developer Console.

## Continuing Slack Setup

You'll activate OAuth, Event Requests URLs (supplied by Dialogflow), and Event Subscriptions now that you've set up Slack and Dialogflow.

**Step8.** Copy ‘OAuth URL’ from Dialogflow.

**Slack**

- In the Slack app settings, go to 'Manage Distribution' and add the bot to your Slack team using the 'Add to Slack' button.
- To use [rich messages](#), copy the 'Events Request URL' value below and paste it into the 'Request URL' field from the Interactive Messages section of your Slack app settings.

More in the [documentation](#).

Slack Client ID \*

Slack Client Secret \*

Slack Verification Token \*

Success page (Optional)

Process all messages

Show URLs for legacy Slack workflow

Slack has released a new OAuth workflow. You can check [this documentation](#) for more information. If you are still using their legacy workflow, please switch to our V1 authentication endpoint.

OAuth URL: <https://dialogflow>

Events Request URL:

Active environment: Draft [?](#)

CLOSE STOP

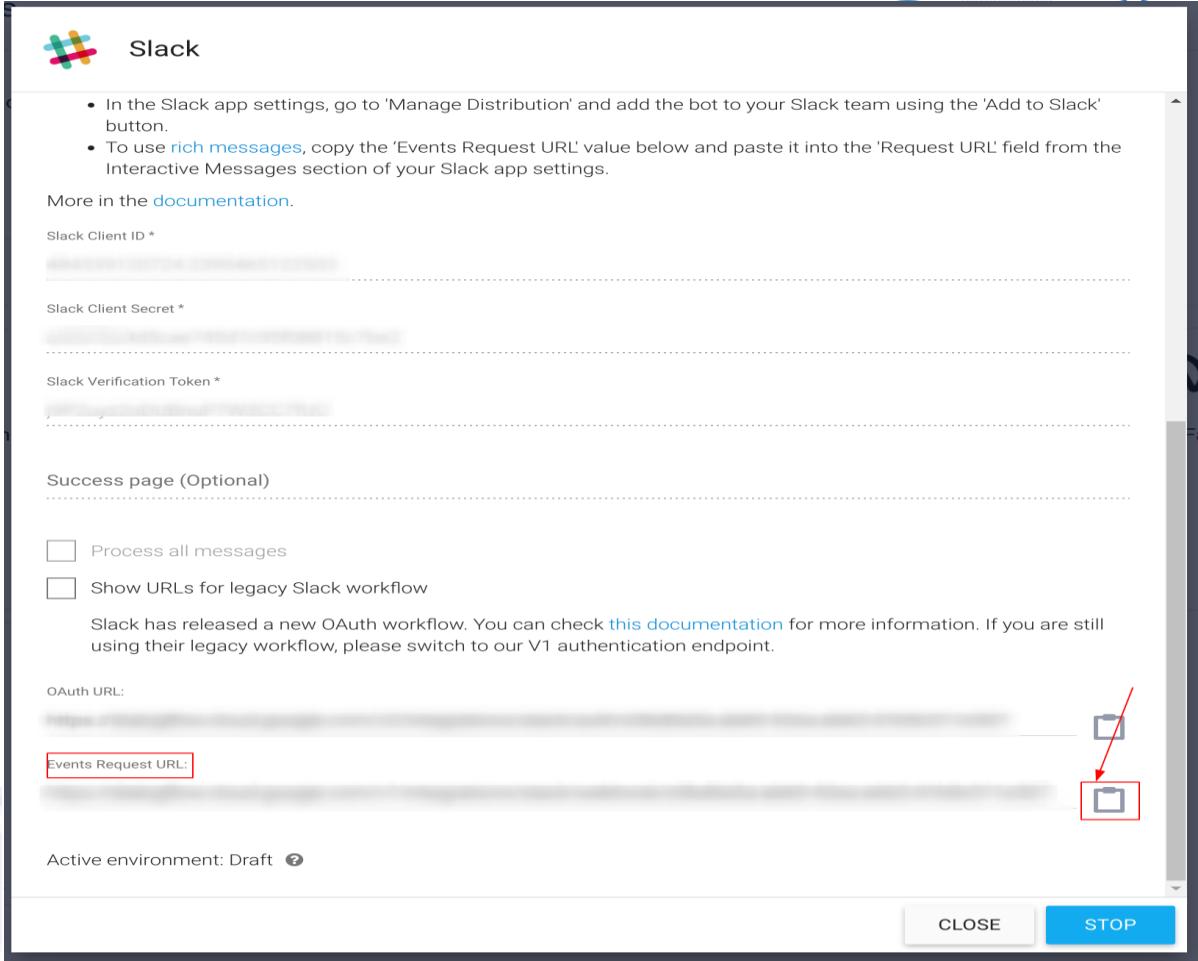
Now, we have to go Slack Development Console and select 'OAuth and Permissions' on the left-hand side.

The screenshot shows the 'OAuth & Permissions' section of the Slack app configuration. It includes a 'Bot User OAuth token' field with a copy button, an 'Access Level: Workspace' note, and a 'Reinstall to Workspace' button. Below this is the 'Redirect URLs' section, which contains instructions about generating Add to Slack buttons and distributing apps. It features an 'Add New Redirect URL' button (marked with a circled 1) and a 'Save URLs' button (marked with a circled 2).

1. Click on 'Add New Redirect URL' and paste the 'OAuth URL' which we had taken from Dialogflow then click on 'Save Urls'.
2. Click on 'Save URLs'.

## Step8. Add Event Request URL and Enable Event Subscriptions

Copy 'Events Request URL' from Dialogflow.



The screenshot shows the Slack Development Console configuration page for an app named "inblog". The page has a header with the Slack logo and a sub-header "Slack". Below this, there are instructions for adding the bot to a Slack team:

- In the Slack app settings, go to 'Manage Distribution' and add the bot to your Slack team using the 'Add to Slack' button.
- To use rich messages, copy the 'Events Request URL' value below and paste it into the 'Request URL' field from the Interactive Messages section of your Slack app settings.

Below the instructions, there's a link to "More in the documentation." and fields for "Slack Client ID \*", "Slack Client Secret \*", and "Slack Verification Token \*".

Under "Success page (Optional)", there are two checkboxes:

- Process all messages
- Show URLs for legacy Slack workflow

A note states: "Slack has released a new OAuth workflow. You can check [this documentation](#) for more information. If you are still using their legacy workflow, please switch to our V1 authentication endpoint."

The "OAuth URL:" field contains a placeholder "Events Request URL:" with a red box around it. To the right of the URL field, there are three small icons: a square, a circle, and a triangle, with a red arrow pointing to the triangle icon.

At the bottom, it says "Active environment: Draft" with a question mark icon. There are "CLOSE" and "STOP" buttons at the bottom right.

Go to Slack Development Console and select '**Event Subscriptions**' which is on the left-hand side.

## Event Subscriptions

### Enable Events

Your app can subscribe to be notified of events in Slack (for example, when a user adds a reaction or creates a file) at a URL you choose. [Learn more.](#)

Request URL Verified ✓

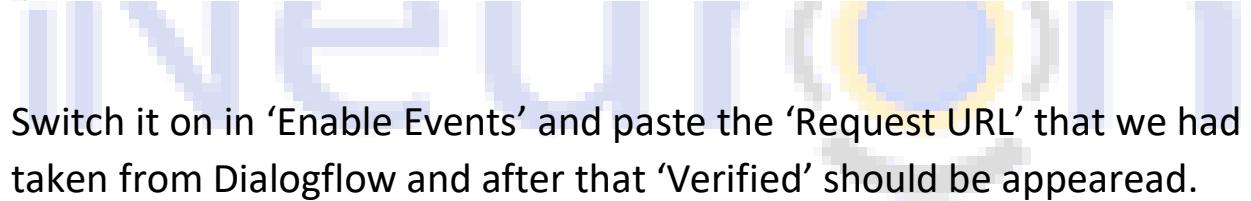
 <https://hooks.slack.com/services/T00000000/B00000000/XXXXXXXXXX> [Change](#)

We'll send HTTP POST requests to this URL when events occur. As soon as you enter a URL, we'll send a request with a [challenge](#) parameter, and your endpoint must respond with the challenge value. [Learn more.](#)

### New event authorization format

 **Recent changes to Events API payloads**

The Events API now sends information about authorized users and workspaces in a new, compact format. [Learn more.](#)

  
Switch it on in 'Enable Events' and paste the 'Request URL' that we had taken from Dialogflow and after that 'Verified' should be appeared.

Now, In '**Event Subscriptions**' we have '**Subscribe to bot events**' click on the '**Add Bot User Event**' and select the '**Event Name**' which we had selected in the image and finally click on '**Save Changes**' button.

Syntax Commands

- Workflow Steps
- OAuth & Permissions
- Event Subscriptions**
- User ID Translation
- Beta Features
- Where's Bot User (?)

**Submit to App Directory**

- Review & Submit

**Slack ❤️**

- Help
- Contact
- Policies
- Our Blog

**Recent changes to Events API payloads**  
The Events API now sends information about authorized users and workspaces in a new, compact format. [Learn more.](#)

**Subscribe to bot events**

Apps can subscribe to receive events the bot user has access to (like new messages in a channel). If you add an event here, we'll add the necessary [OAuth scope](#) for you.

Event Name	Description	Required Scope
<a href="#">im_history_changed</a>	Bulk updates were made to a DM's history	im:history <span style="font-size: small;">Delete</span>
<a href="#">message.channels</a>	A message was posted to a channel	channels:history <span style="font-size: small;">Delete</span>
<a href="#">message.groups</a>	A message was posted to a private channel	groups:history <span style="font-size: small;">Delete</span>
<a href="#">message.im</a>	A message was posted in a direct message channel	im:history <span style="font-size: small;">Delete</span>

**Add Bot User Event**

Discard Changes Save Changes

## Step9. Now, add Slack bot to a team.

In Slack Developer Console, select ‘Manage Distribution’ and under that click on the ‘Add to Slack’ button.

**Manage Distribution**

**Settings**

- Basic Information
- Collaborators
- Socket Mode
- Install Your App
- App Manifest BETA
- Manage Distribution**

**Features**

- App Home
- Org Level Apps
- Incoming Webhooks

**Share Your App with Your Workspace**

You can use the URL and the Add to Slack button below to share your app with the Xyz Marketing workspace. Activate distribution (below) to share your app with any workspace.

**Embeddable Slack Button**

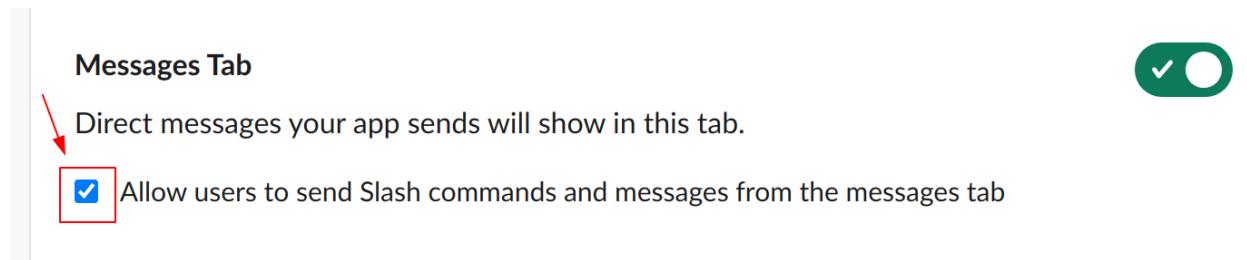
[Add to Slack](#)

<a href="https://slack.com/oauth/v2/authorize?client\_id=4843391" style="color: inherit; text-decoration: none;">Copy

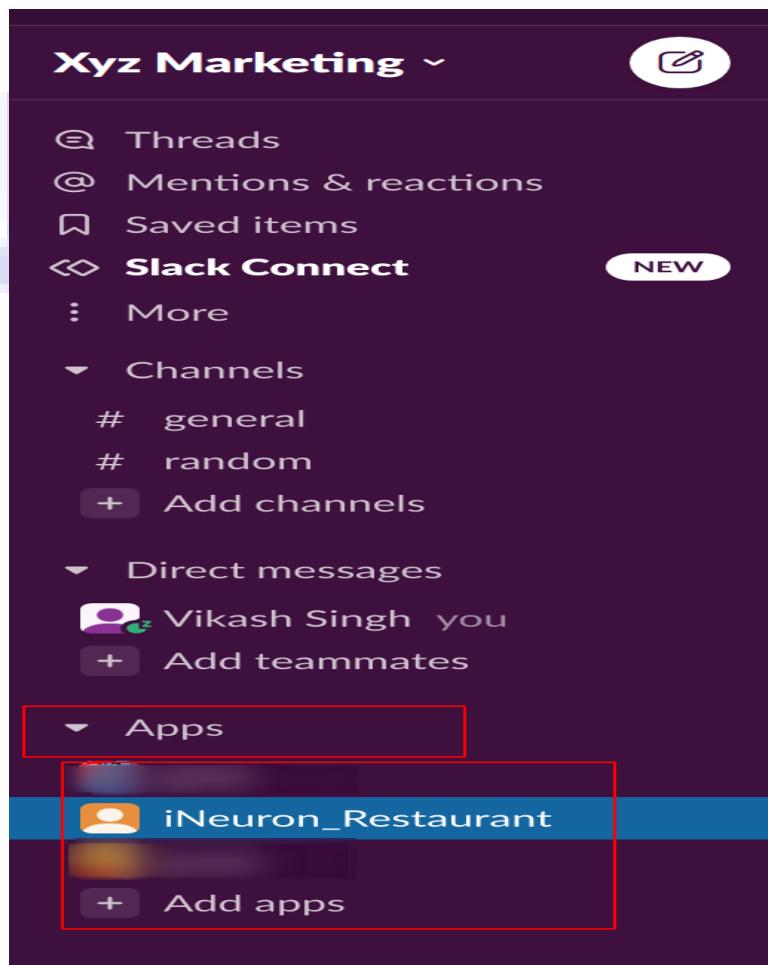
**Sharable URL**

https://slack.com/oauth/v2/authorize?client\_id=484339120724.2395465122503&scop... [Copy](#)

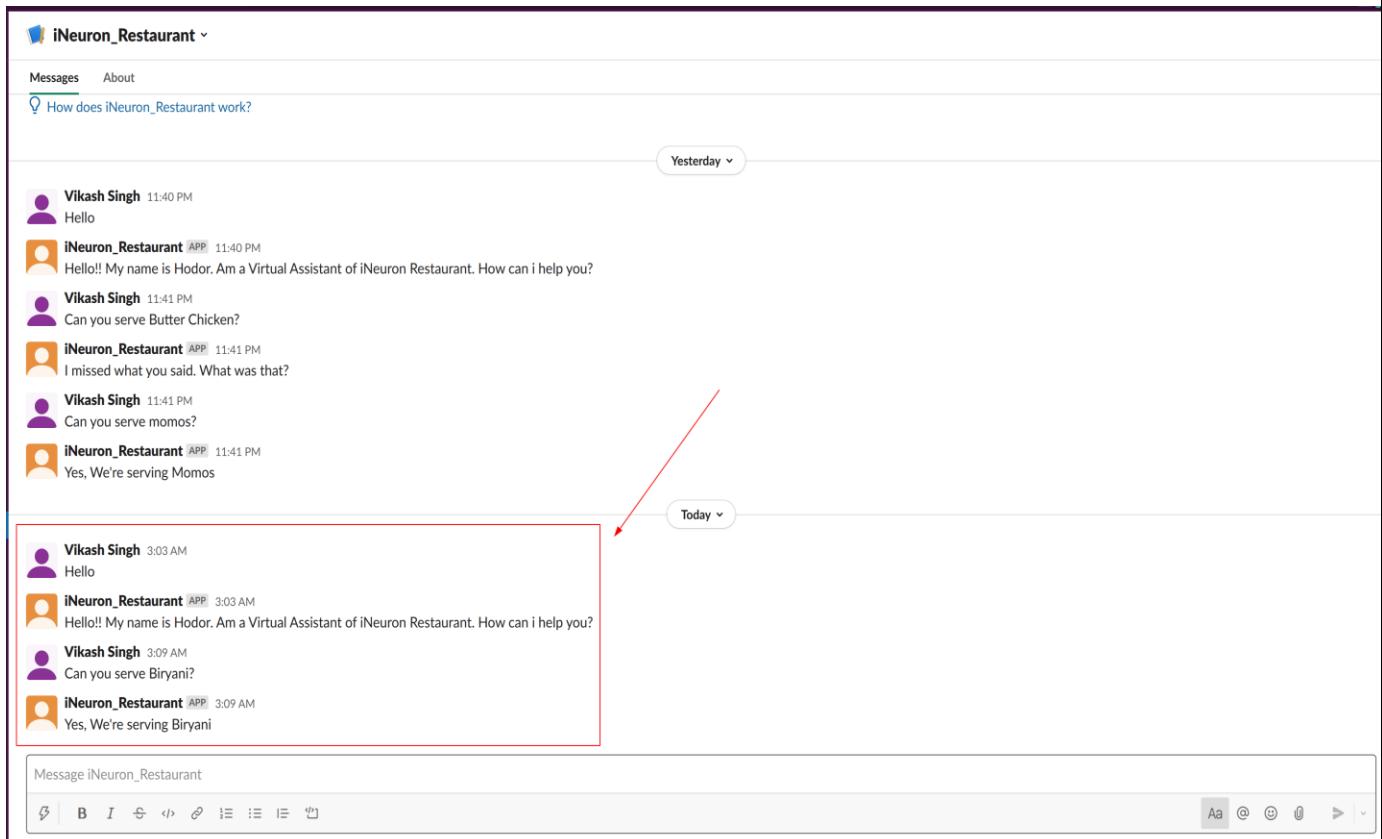
**Note:** In Slack Development Console, select 'App Home' which is in the left-hand side menu. Don't forget to select the check box which is present under 'Messages Tab'.



Go to your slack app or page and you can check on the left-hand side under 'Apps' section, we have our app available.



Now, will check our app is working or not.



Yes, from above image we got that, our Slack Bot is working perfectly.