



Build a ticket booking Bot using Google's Dialogflow

Hands-On Workshop | Digital Summit '18

Miracle Innovation Labs
Miracle Software Systems, Inc.

Build a ticket booking Bot using Google's Dialogflow

Introduction

This document contains the step-by-step process of creating a chat bot with Google's Dialogflow (before the name used to be API.ai) NLP and will teach you how to create a bot and integrate the Chatbot with Facebook Messenger using one click integration in Dialogflow.

This guide was prepared by [Miracle's Innovation Labs](#)

Pre-Requisites

All attendees must have their workstation (with Internet) to participate in the lab (both PC and Mac are compatible). The following pre-requisites will help you to make the Hands-on Lab experience easier.

- A Google account is required for Dialogflow
- Access for Facebook and Facebook for Developers
- Facebook account to create a page for Chat Bot integration
- Download and install Node JS and ngrok
- Text Editor such as Visual Studio (or) Notepad++

Technology Involved

- NLP - Google's Dialogflow
- Server Side - Node JS

Labs Steps

So, let us get started with the bot!

The following steps will outline how you can create an Ticket Booking Bot using Google's Dialogflow and integrate it with Facebook Messenger. Users will be able to directly message your bot through Facebook Messenger to book your flight ticket.

Step #1 | Create Dialog Model in Dialogflow

Create an account for Dialogflow. Go to <https://dialogflow.com/> and click on **GO TO CONSOLE** or **SIGN UP FOR FREE**.

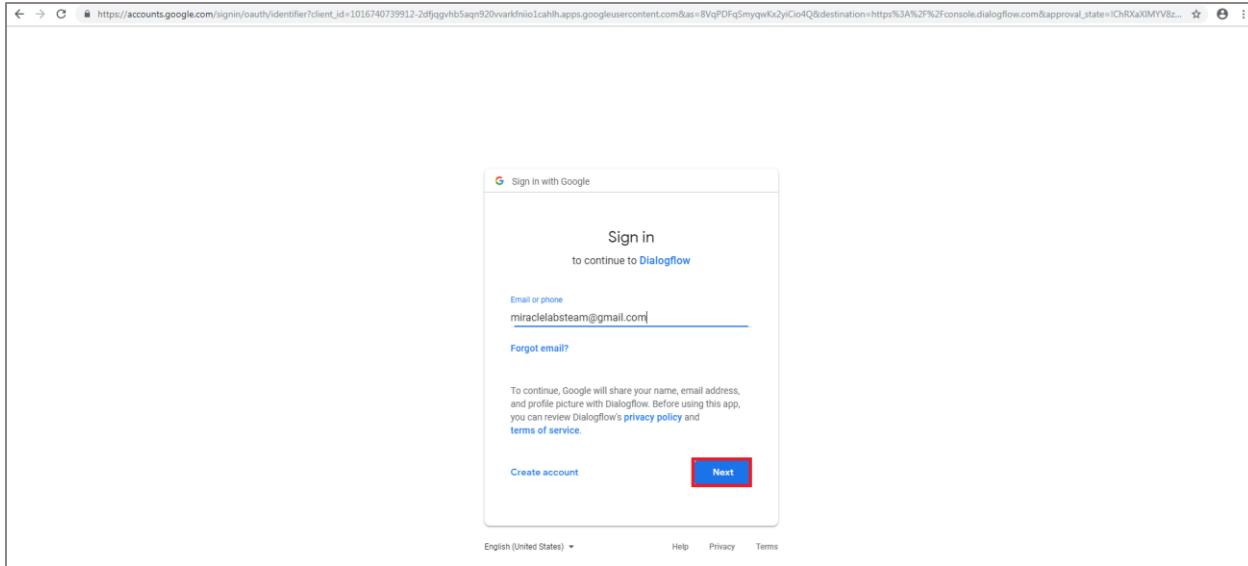
The screenshot shows the official Dialogflow website at <https://dialogflow.com/>. At the top, there's a navigation bar with links for Overview, Case studies, Docs, Blog, Pricing, and Support. A red button labeled "GO TO CONSOLE" is prominently displayed. Below the navigation, a large banner titled "Intro to Dialogflow" features a woman working on a laptop with a city skyline in the background. The banner includes a "Watch later" and "Share" button. To the left of the banner, there's a section titled "Build natural and rich conversational experiences" with a "SIGN UP FOR FREE" button. Below this, there are three columns of text: "Powered by Google's machine learning", "Built on Google infrastructure", and "Optimized for the Google Assistant". A testimonial from "Mandi Galluch, Digital Experience Program Leader, Domino's" is highlighted in a teal box: "Dialogflow is user-friendly, intuitive, and just makes sense. Its natural language processing (NLP) is the best we've tried."

After clicking on any one of the button as shown above, it will be redirected to the console of Dialogflow for login. Now, click on **Google** button.

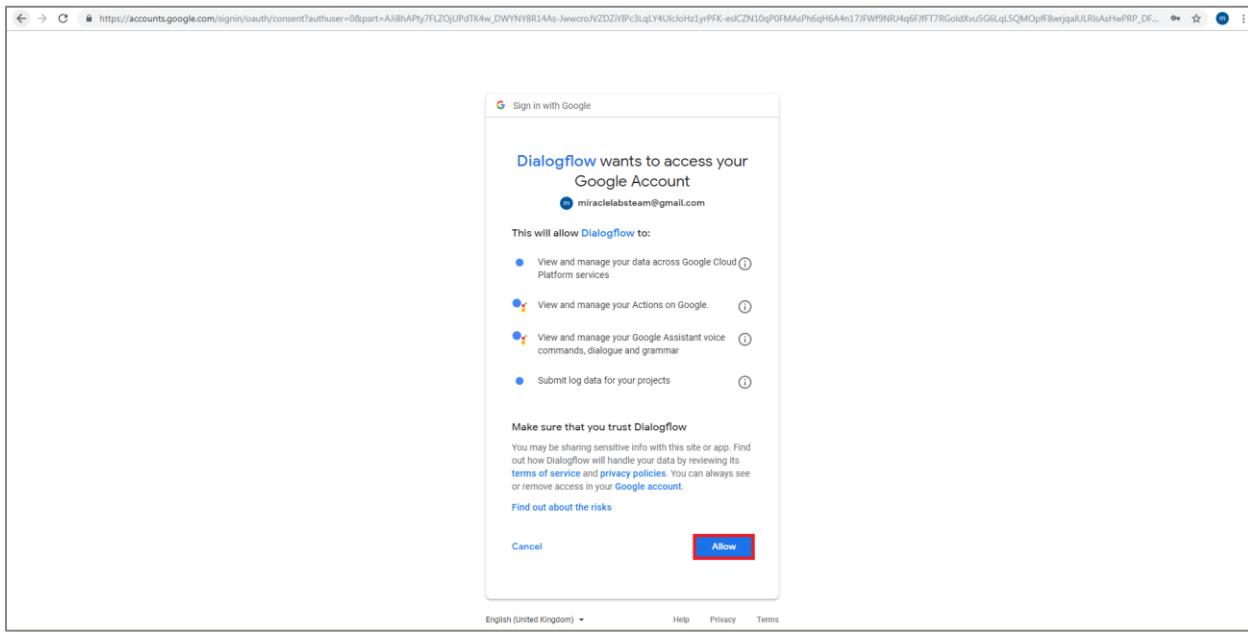
The screenshot shows the "Sign in with" page of the Dialogflow API Client at <https://console.dialogflow.com/api-client/#/login>. It features a "Google" button with a red border. Below the button, a note states: "In order to improve the system security, we are migrating authorization exclusively to Google accounts." There are also links for "Terms of use and privacy policy" and "BACK TO DIALOGFLOW".

Dialogflow account can be created with Google account credentials. If you have a Google account then use those credentials and authorize yourself.

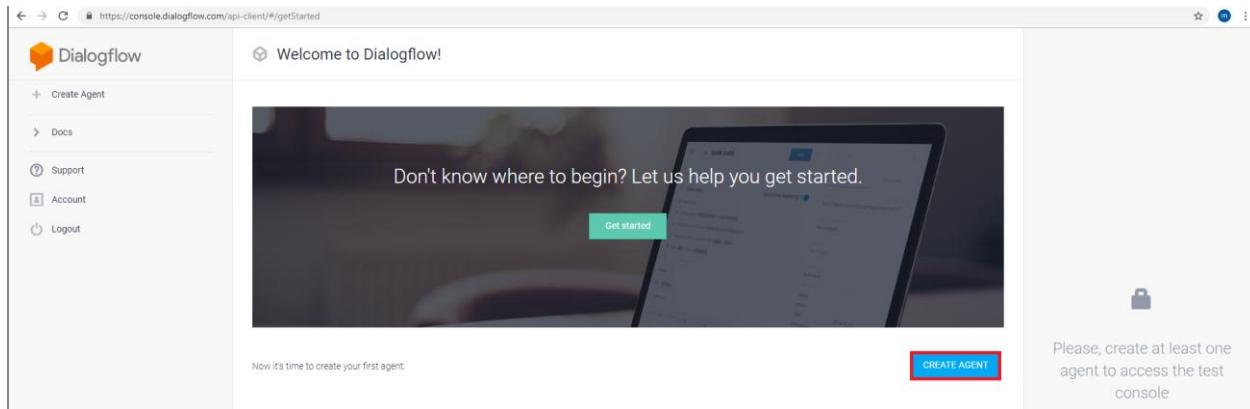
If you don't have a Google account, create by clicking the below link,
<https://accounts.google.com/SignUpWithoutGmail?hl=en/>



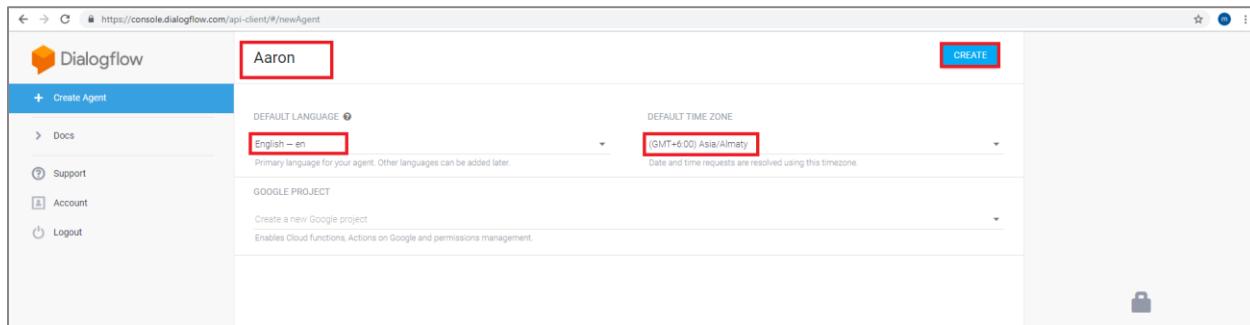
Once you get signed in with your credentials, you will get the permission for accessing Dialogflow with your Google Account. To provide access click on **Allow** as shown below,



Now, you will get a Dialogflow home page where you have to create an agent/bot. Click on **CREATE AGENT** (if a pop up opens stating don't have correct permissions just Authorize).

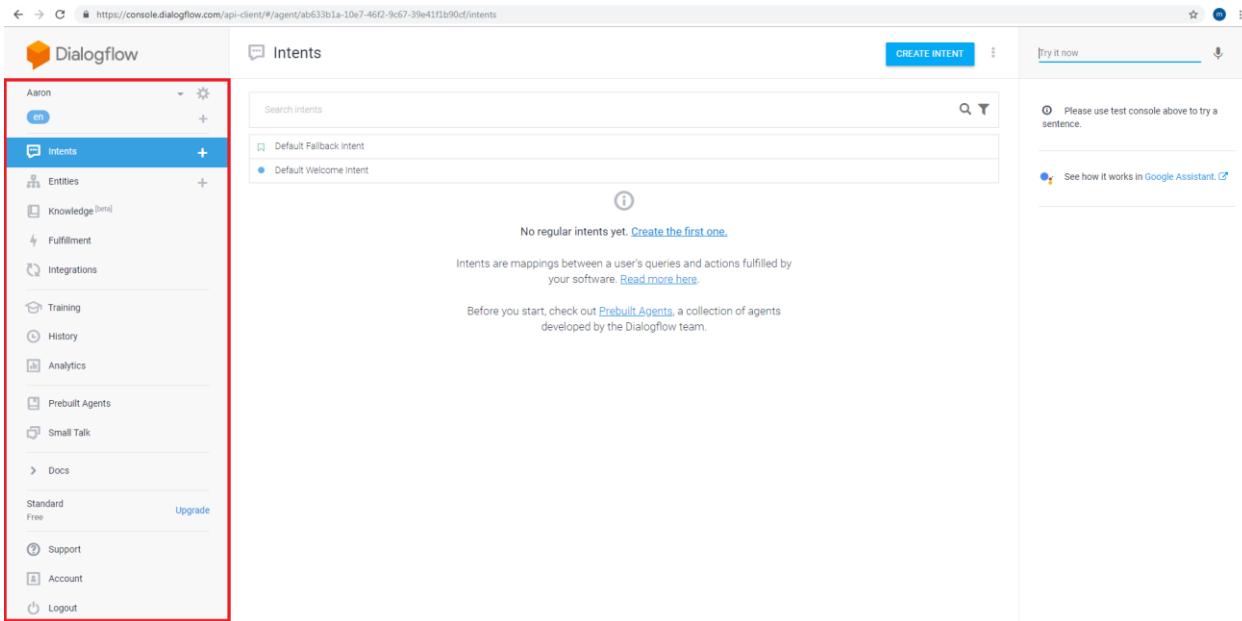


Once you click on **CREATE AGENT**, then you need to provide an agent name (In our case it is **Aaron**, choose the language, and time zone from the drop down respectively and click on **CREATE** button.



Once you are done with the creation of your agent you will get the dashboard with lot of different options on the left hand side which are used for bot creation. You will also have two Default Intents initially,

- **Default Welcome Intent** - This Intent will initially respond to the users, when users engage to the service
- **Default Fallback Intent** - This Intent responds to the user, when input is not recognized

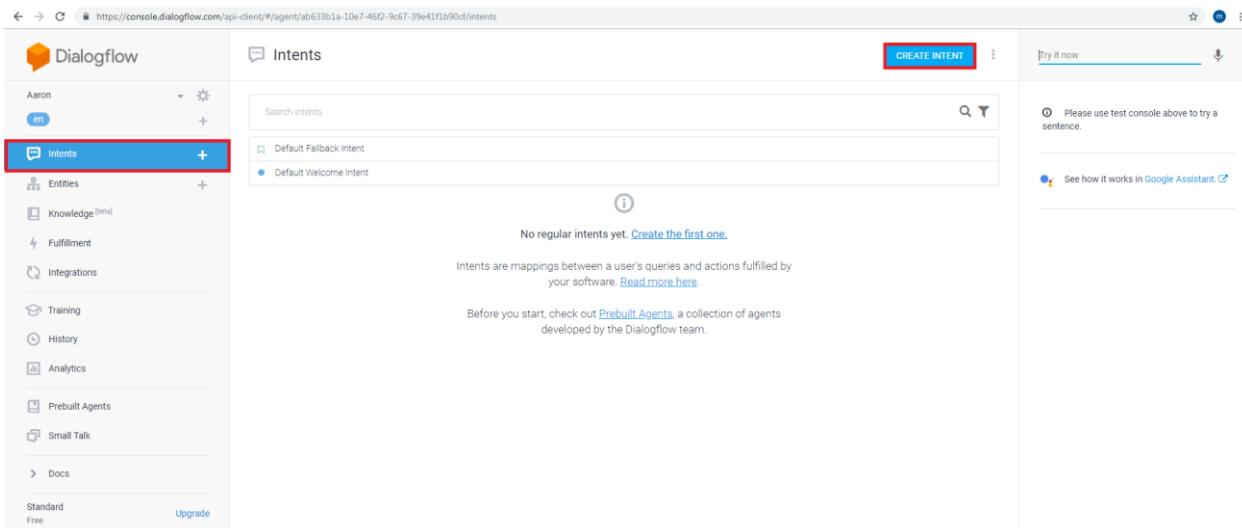


You can read about these in detail through [Dialogflow docs](#).

By now, we have a basic setup of Dialogflow. Now you have to create Intent, Entities and Text response for bot. Let's see one by one.

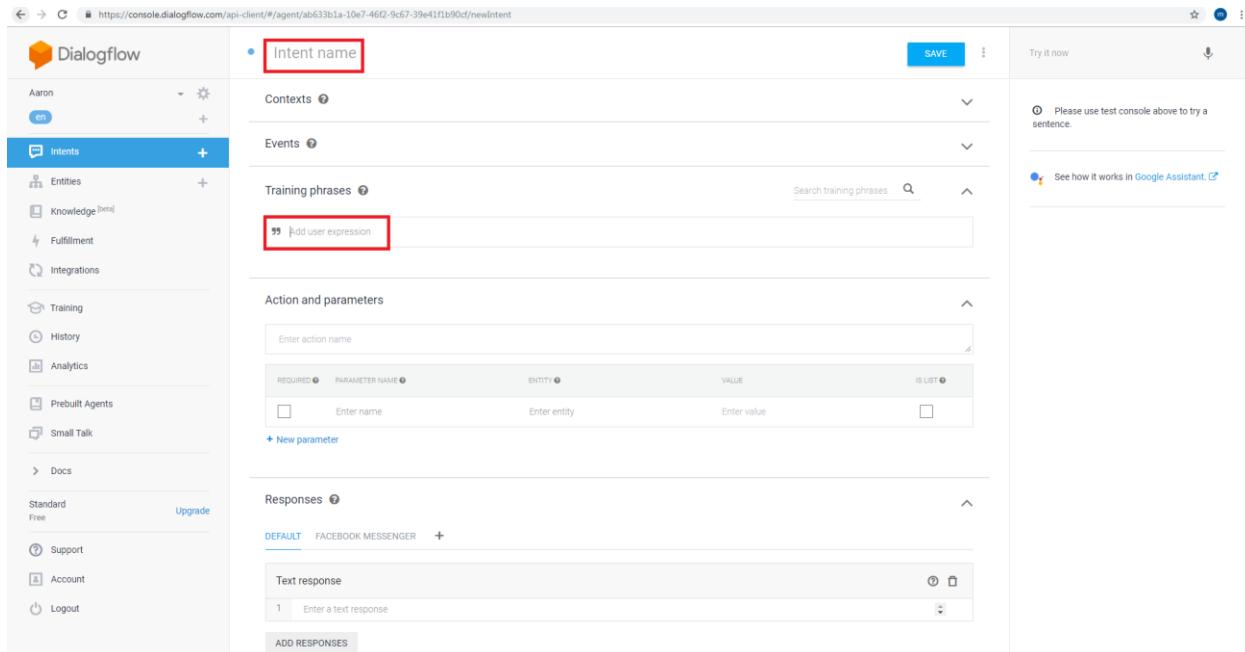
Creation of Intent

Click on "+" sign on Intents tab on left hand side or click on **CREATE INTENT** button for creating new Intent.



Now you need to provide the **Intent name** and **Add user expression** under **User says** section for creating the Intent.

Note - Under User says, as many expressions you provide the Agent will become that matured.



Now, provide the response in the **Text response** section which is used to respond back to the intent when it detects same kind of user examples. We can provide multiple responses for the same Intent. Once it is done, click on **SAVE** for saving the created intent.

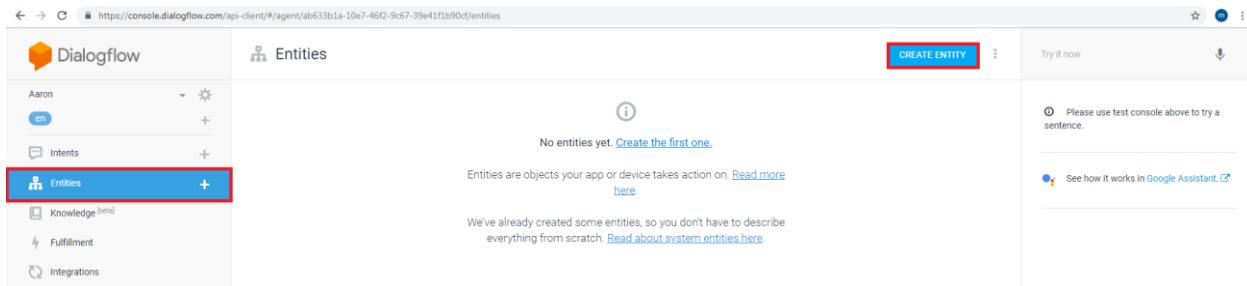
The screenshot shows the Dialogflow interface for creating a new intent. On the left, there's a sidebar with various options like Entities, Knowledge, Fulfillment, Integrations, Training, History, Analytics, Prebuilt Agents, Small Talk, Docs, Support, Account, and Logout. The main area is titled 'Intent name' and has sections for Contexts, Events, Training phrases, Action and parameters, and Responses. Under 'Responses', there's a 'Text response' tab selected, which contains a text input field with placeholder 'Enter a text response'. A red box highlights this text input field.

Once you provide the response in the Text response section, if your responses from all the intents need to send to the Facebook channel, go to every intent and click on **FACEBOOK MESSENGER** and enable the button to use your text responses as default responses as shown below.

This screenshot shows the same Dialogflow interface as the previous one, but with a different focus. The 'Facebook Messenger' tab under the 'Responses' section is selected, indicated by a red box. Below it, there's a note: 'Response from this tab will be sent to the Facebook Messenger integration. Use response from the DEFAULT tab as the first response.' To the right of this note is a blue toggle switch, which is also highlighted with a red box. Other tabs like 'Text response' and 'Image response' are visible but not selected.

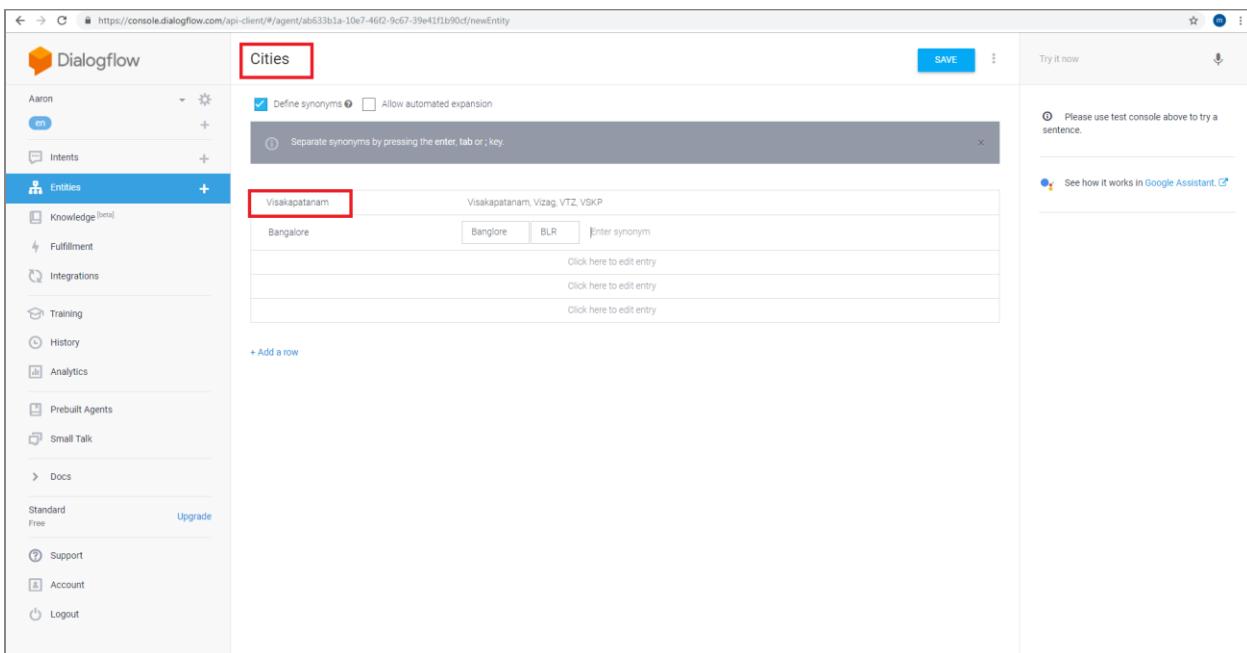
Creation of Entities

Click on "+" sign on Entities tab on left hand side (or) click on **CREATE ENTITY** button, which is at top right corner for creating the Entity.



The screenshot shows the Dialogflow Entities creation interface. On the left sidebar, the 'Entities' tab is selected and highlighted with a red box. At the top right, there is a blue 'CREATE ENTITY' button with a red box around it. The main area displays a message: 'No entities yet. Create the first one.' Below this, there is a note: 'Entities are objects your app or device takes action on. [Read more here](#)'. Another note below it says: 'We've already created some entities, so you don't have to describe everything from scratch. [Read about system entities here](#)'. On the right side, there is a 'Try it now' button and a link to 'See how it works in Google Assistant'.

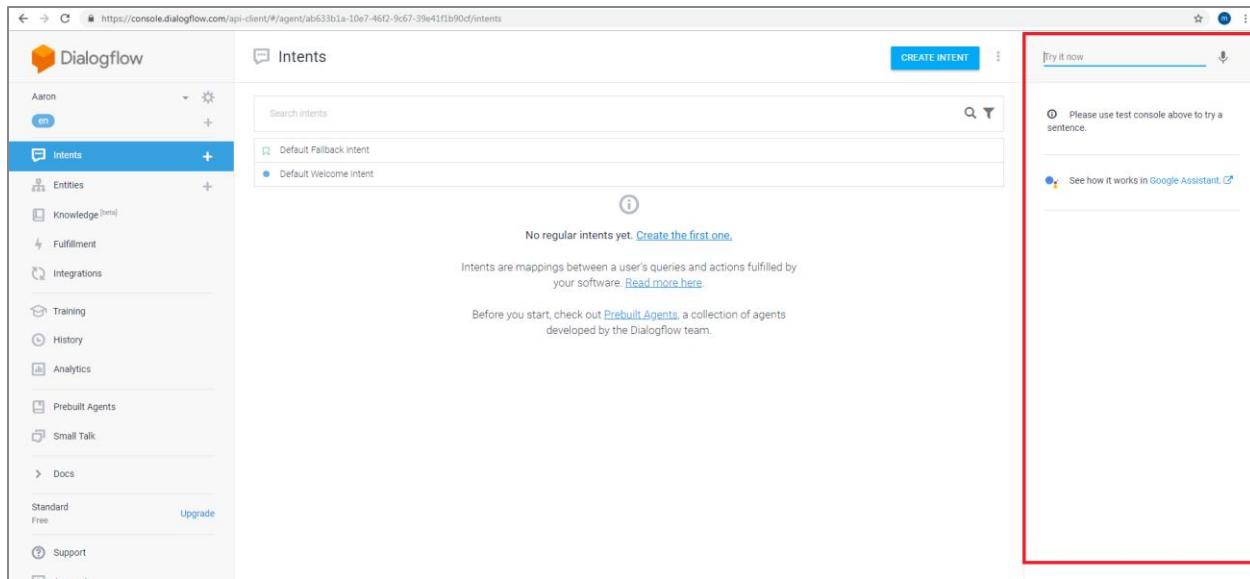
You need to give the entity name and also you have to provide the reference value and synonyms for it while creating the Intent as following,



The screenshot shows the 'Create Entity' form for 'Cities'. The entity name 'Cities' is entered in the input field, which is highlighted with a red box. The 'SAVE' button is located at the top right of the form. The form includes fields for 'Define synonyms' (checkbox checked), 'Allow automated expansion' (checkbox unchecked), and a text input for 'Separate synonyms by pressing the enter, tab or ; key'. A list of synonyms is shown: 'Visakapatnam' (highlighted with a red box), 'Visakapatnam, Vizag, VTZ, VSKP', 'Bangalore', and 'BLR'. There is a 'Enter synonym' button and three empty rows for additional entries. A '+ Add a row' button is at the bottom. The left sidebar shows the 'Entities' tab selected again.

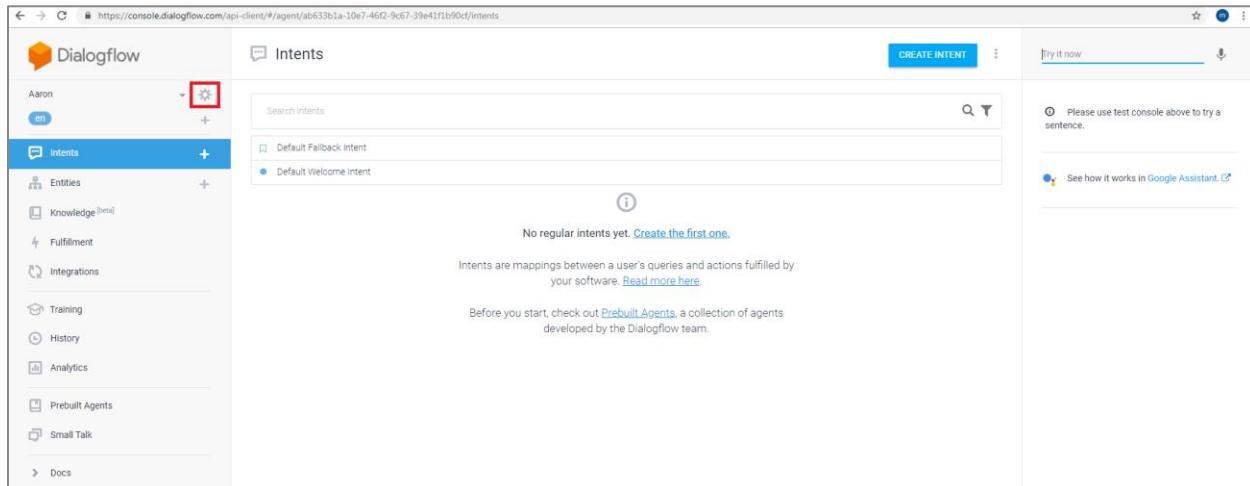
Then click on **SAVE** button on the top right to save the created entity.

You can test your Bot in right side test console provided by Dialogflow,

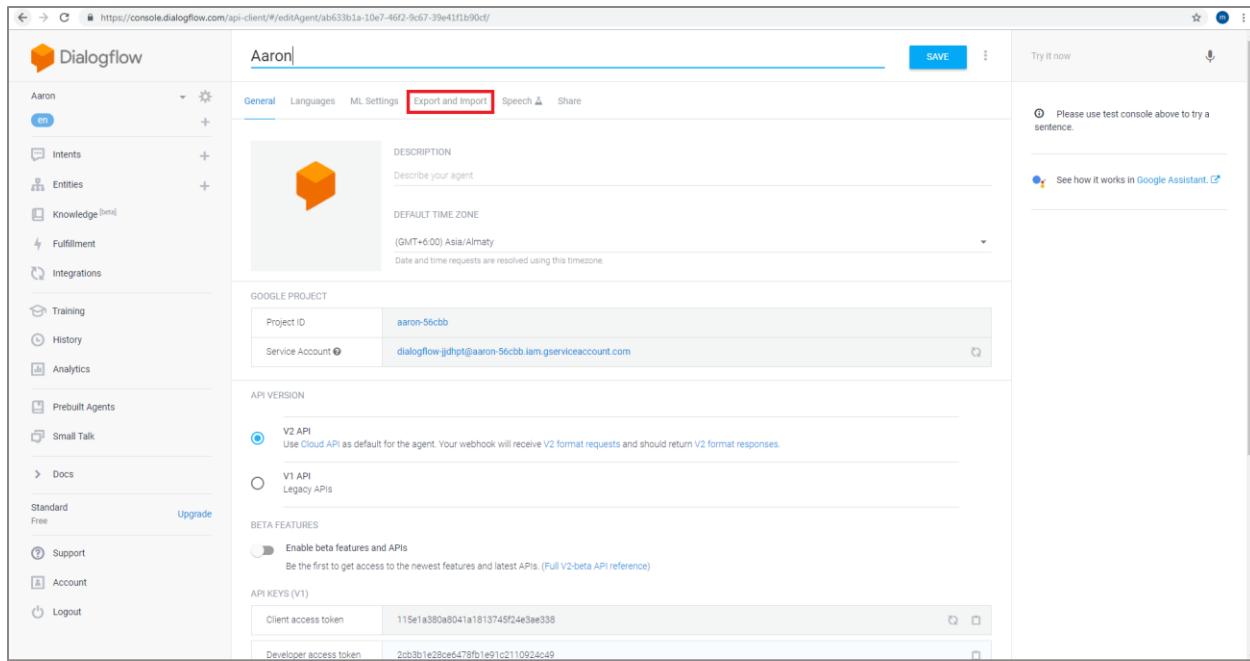


From our GitHub repository, download the zip file in your system. You can find the sample Agent which is created with the name **dialog_model.zip**

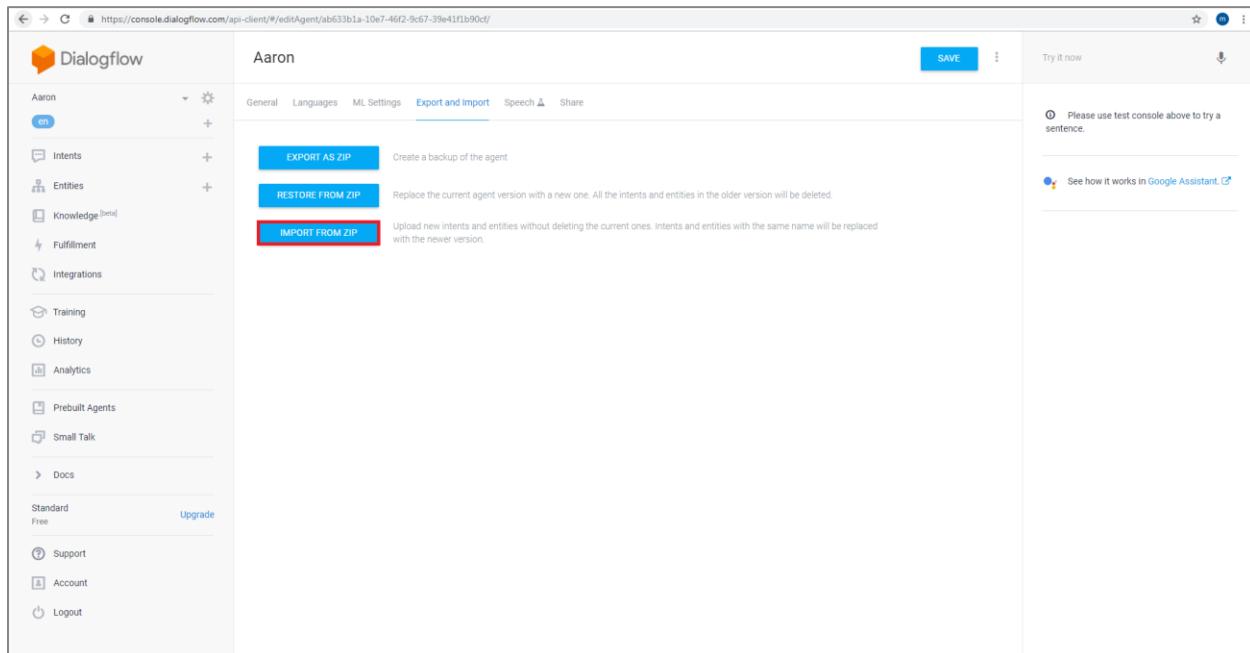
Now go to Dialogflow console and click on gear icon as shown below to import Intents and Entities.



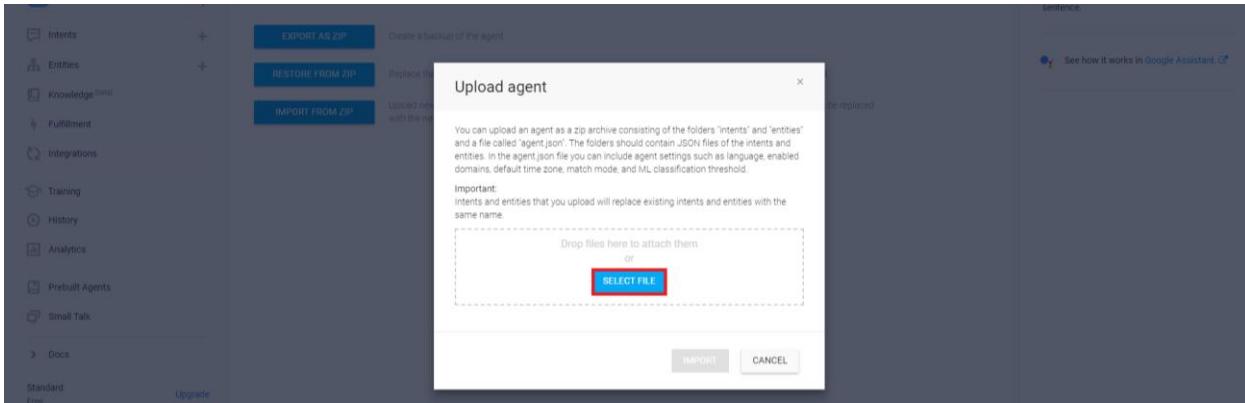
By default, you will be landed on General session page. Select **Export and Import** tab.



Click on **IMPORT FROM ZIP** tile, which will prompt you to select the zip file from your system.

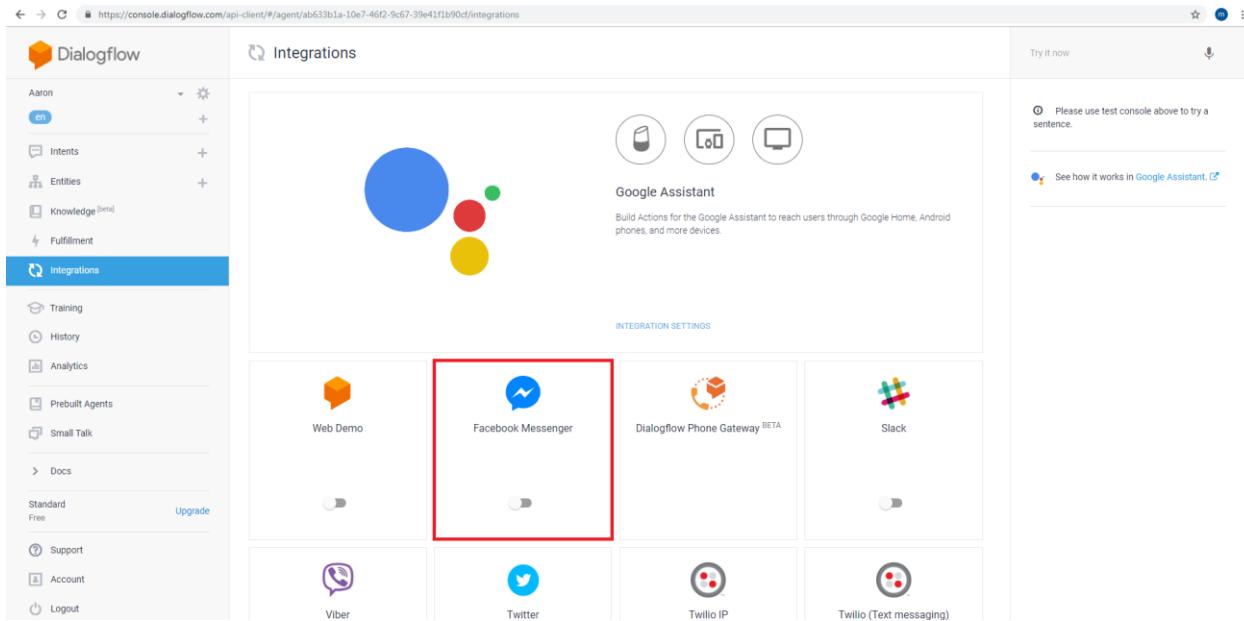


Now you will get a pop-up for asking you to **SELECT FILE**, you have to upload the zip file which you downloaded previously from GitHub repository. Once upload is done click on **IMPORT** button.

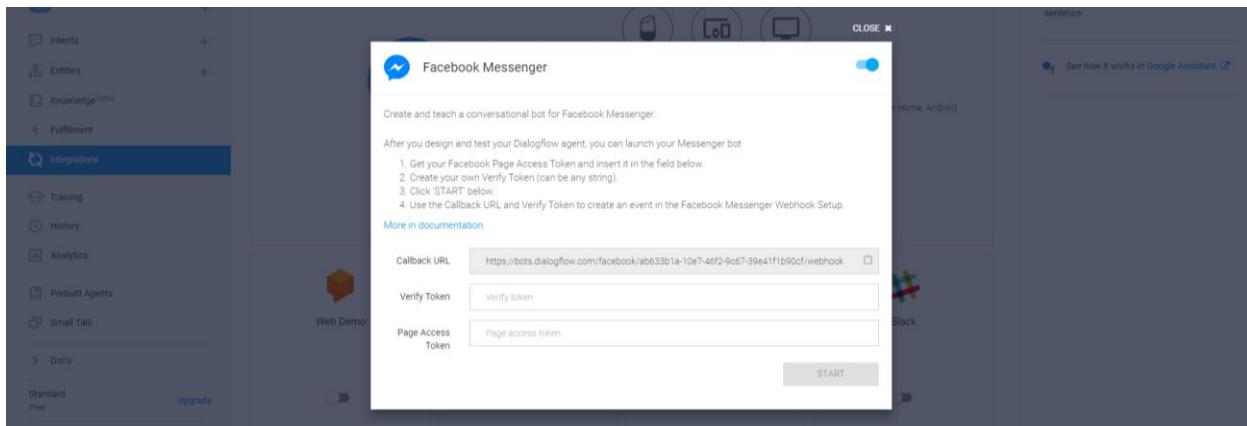


Now, you will find the agent of Aaron Bot which you imported.

In your agent, go to **Integrations** tab on the left side menu in Dialogflow and you will see a bunch of one click integrations. In that, go ahead and enable the Facebook Messenger integration.



Once you click on Facebook Messenger integration, you will get a default Callback URL in a pop-up. Copy that Callback URL.



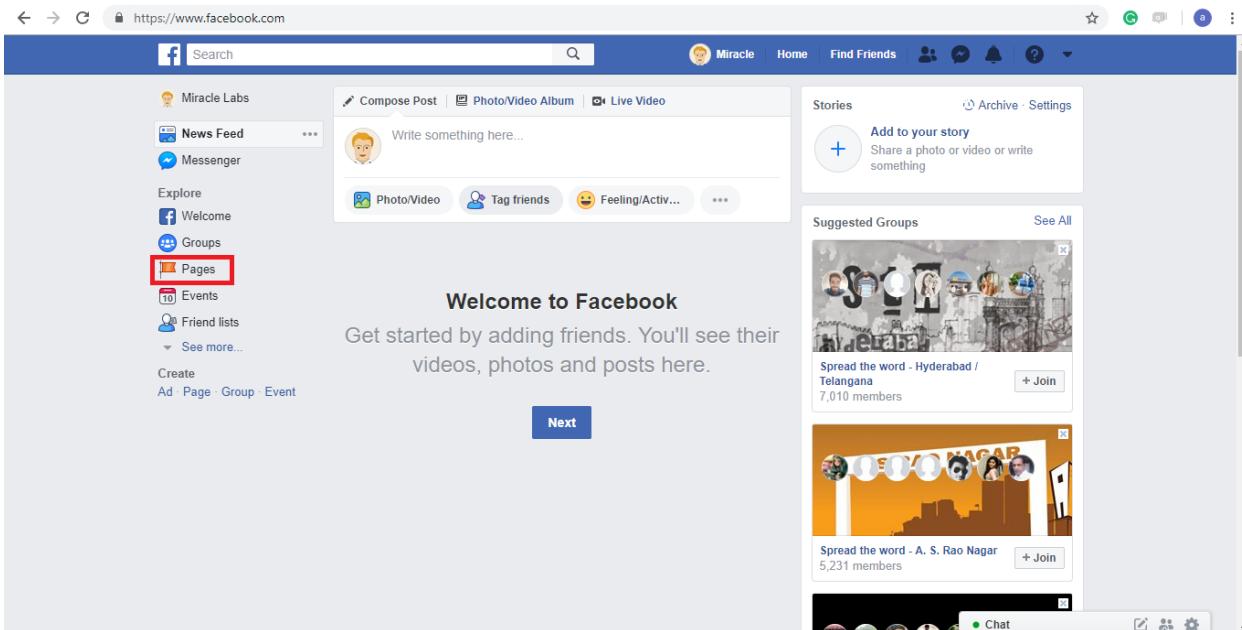
Step #4 | Create Facebook Page to Access Chatbot

The first step will be to create a Facebook account and **Log In** by providing Email or Phone and Password at, <https://www.facebook.com/>

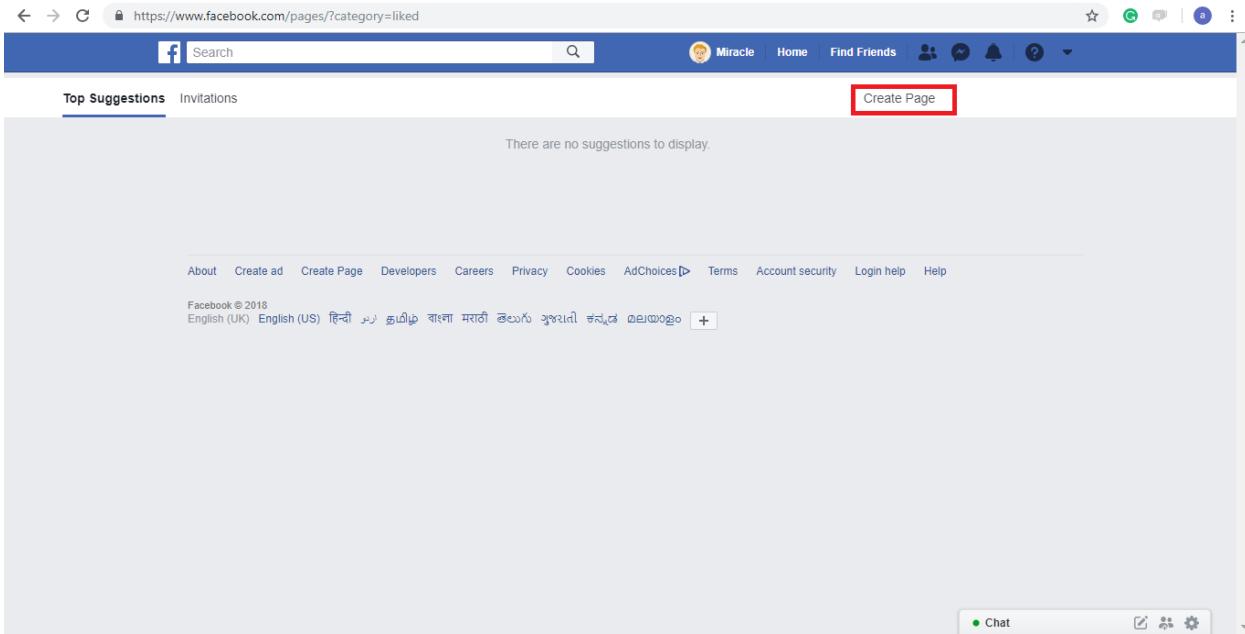


Once you login into Facebook, you need to create a new page by opening the below link in a new tab,

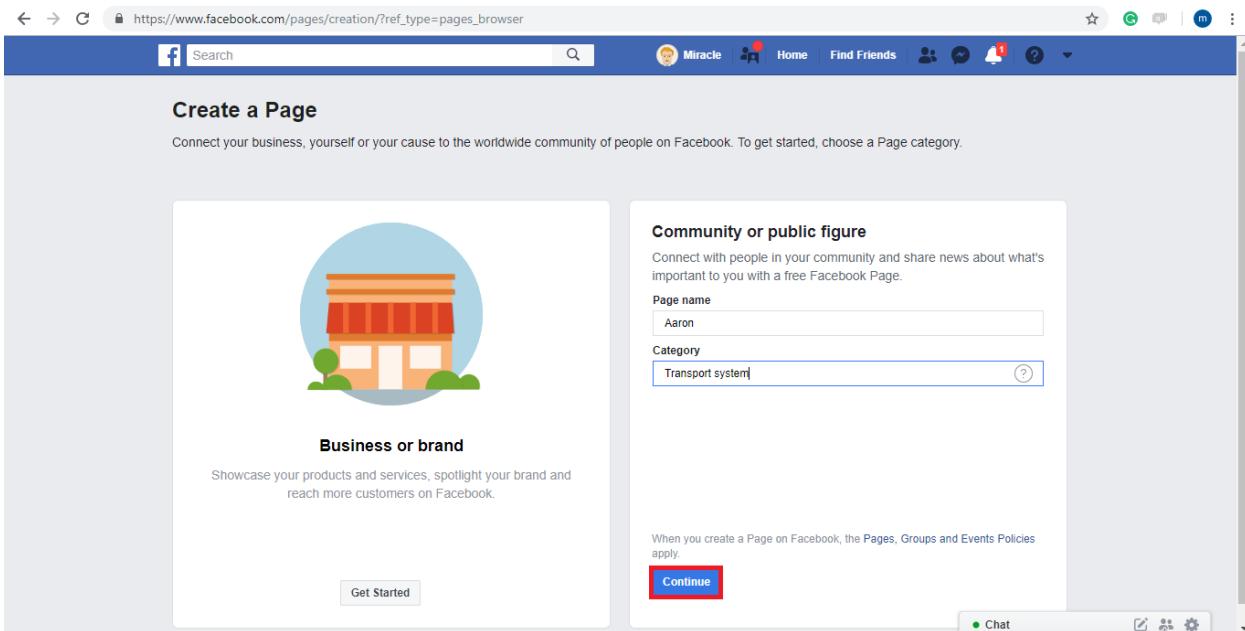
<https://www.facebook.com/pages/create/> or else click on **Pages** as shown below.



If you are creating a page for very first time, you will not find any existing pages. Then click on **Create Page** button as shown below,



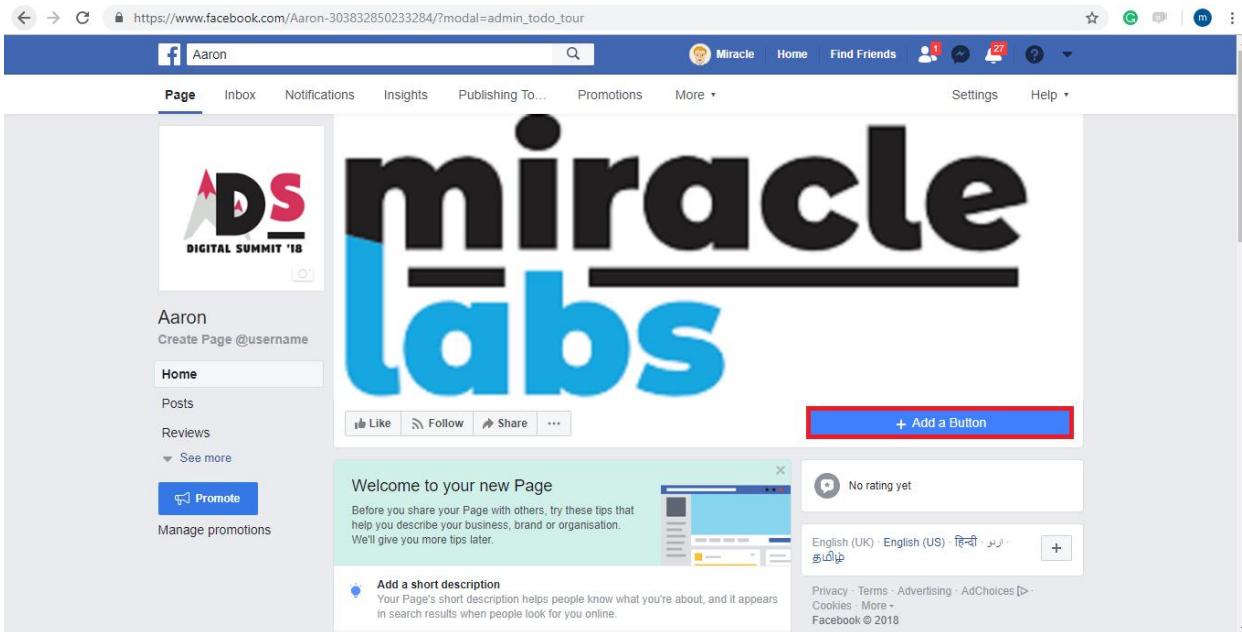
Now, provide a **Page name** and select **Category** of that page and click on **Continue** button. Note that page name should be unique.



After clicking on Continue button, your page will be created and looks like below,

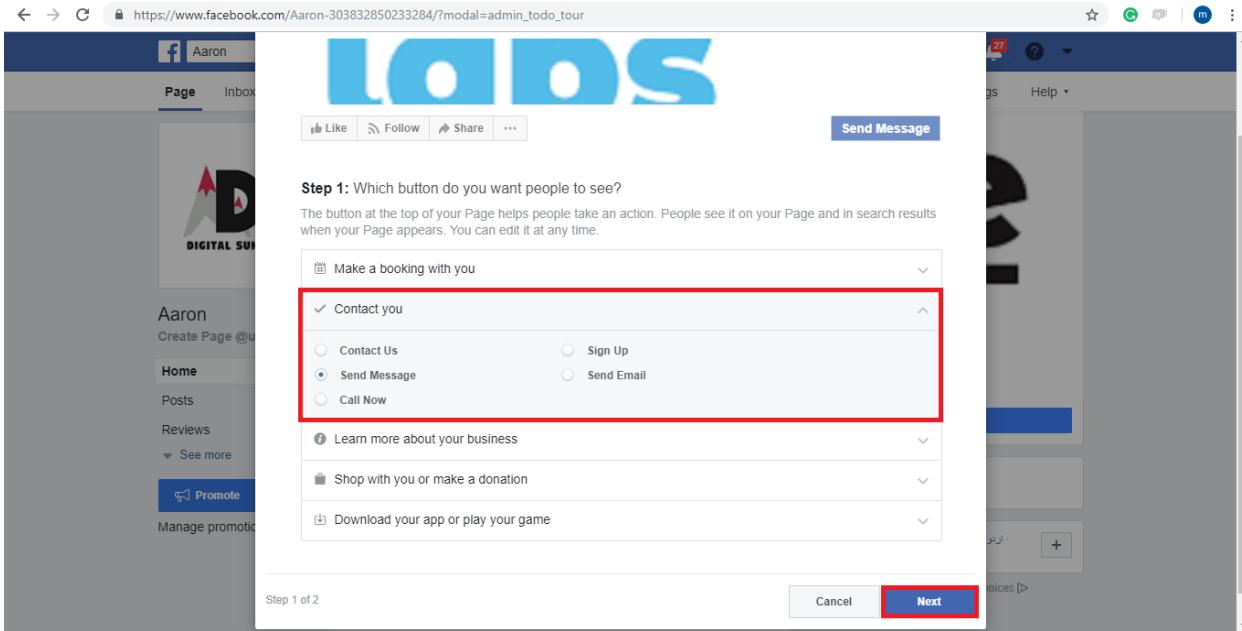


Once you are done with the page creation, click on **+ Add a Button**.

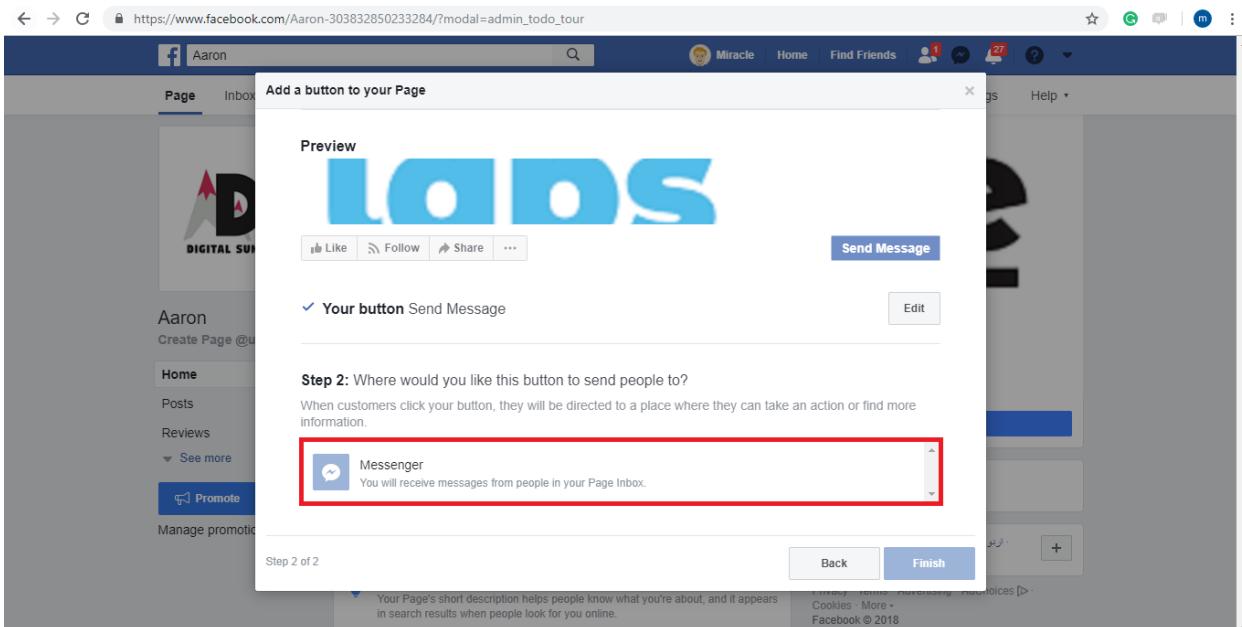


We need to get a Send Message button to start the conversation with bot.

To get that, click on **+ Add a Button**. You will get the options to select the button category. Please select the Contact you option and choose **Send Message** radio button as below and click on **Next**.

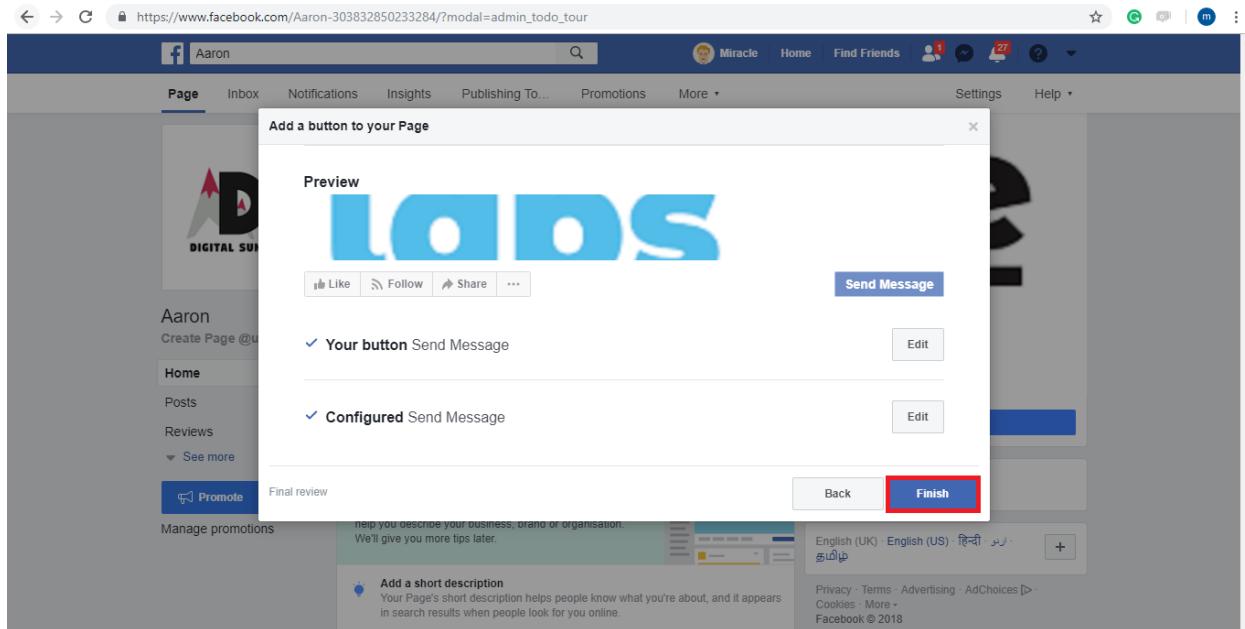


After clicking on **Next** button, it will ask for another option - **Where would you like this button to send people to?** Place your cursor over **Messenger** and click on it.

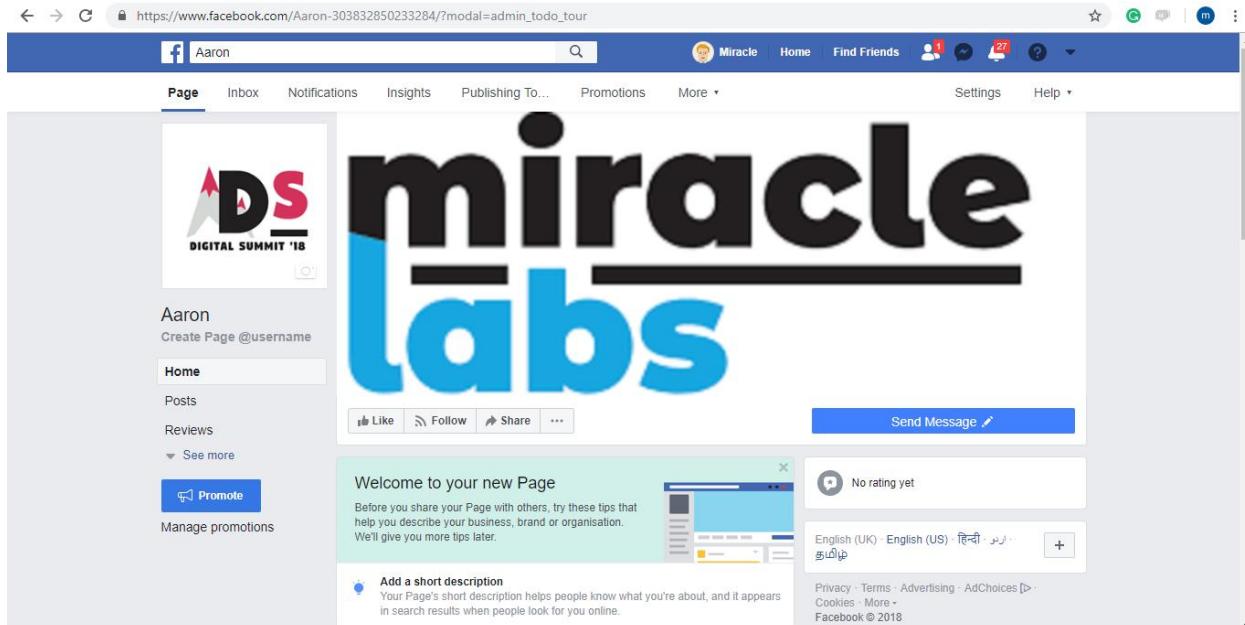


Once you complete the above steps, **Finish** button will be enabled and click on that **Finish** button.

Note: If you want to edit any option that you selected previously, click on Edit button which is available above the **Finish** button.



Now, you can see the **Send Message** button on your page which is replaced with + Add a Button.



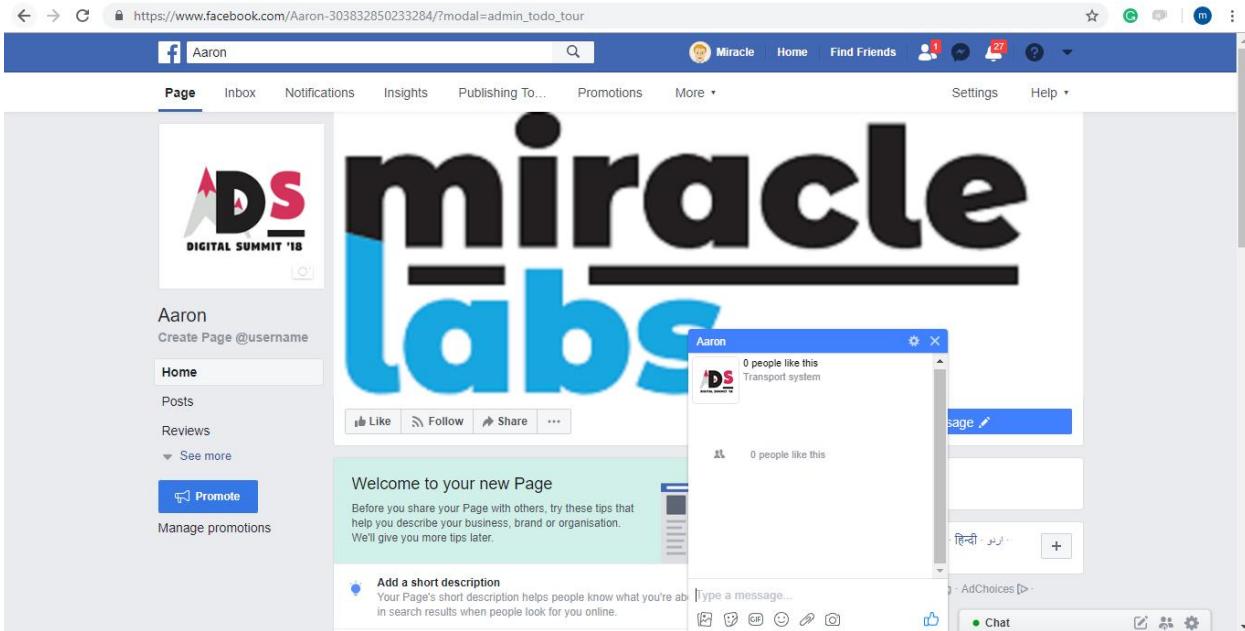
If you want to test the bot that you created, place your mouse cursor over the **Send Message** button.



To start the conversation with chatbot that you created, click on **Test button** from the options.

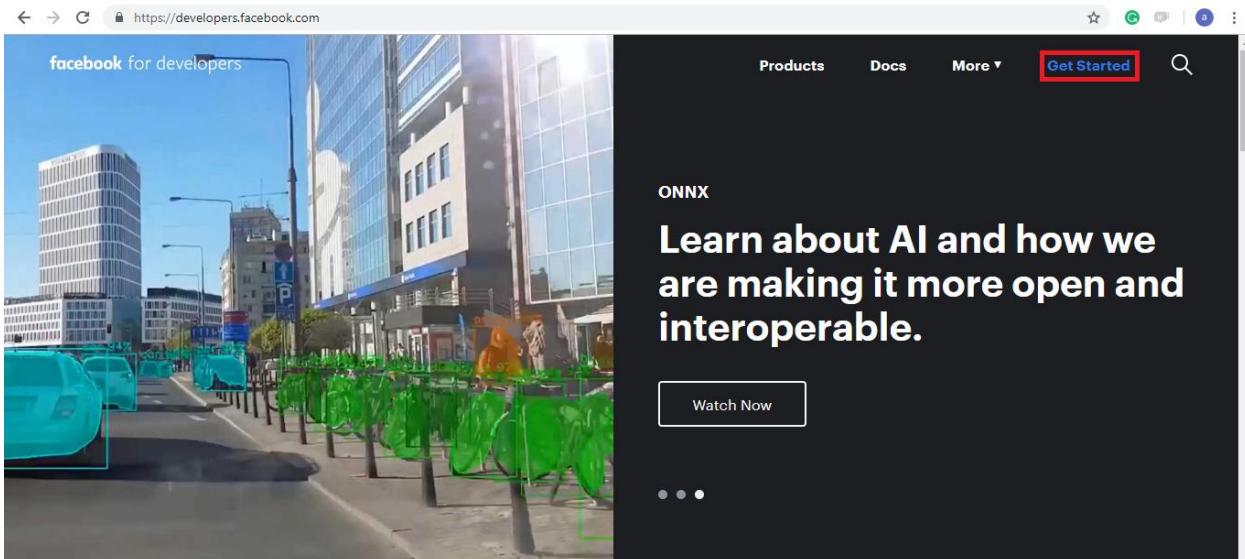


Now, chat window will be opened to start the conversation with your bot.

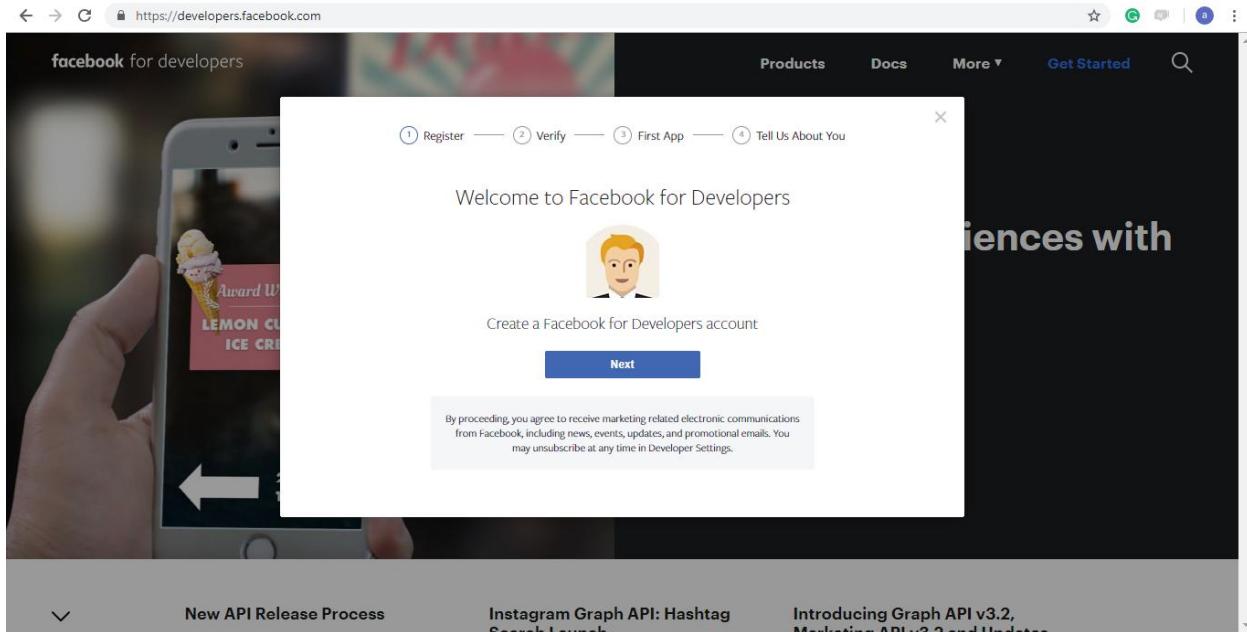


Step #5 | Accessing Facebook for Developers

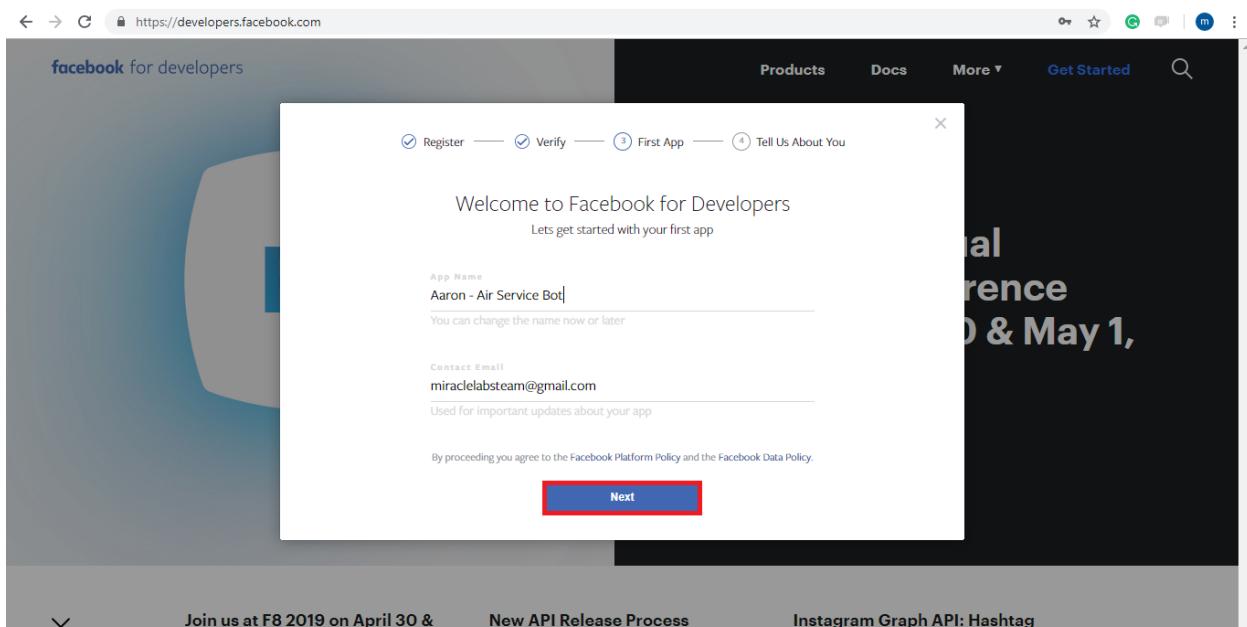
Login into Facebook for Developers, for creating apps in order to generate page access token <http://developers.facebook.com/>. If you are already logged in, click on **Get Started**.



Complete all 4 steps - **Register**, **Verify**, **First App** and **Tell Us About You**. To complete the above steps, authenticate yourself by Email or Phone and provide App name.



By default, it gives one App Name. Specify a display name for your app and then click on **Next**.



Note: If you have already created the app and you want to use it, select it from the **My Apps** menu.

The screenshot shows the Facebook Developers website at <https://developers.facebook.com/apps/337146167016985/add/>. The 'My Apps' tab is selected. The 'PRODUCTS' sidebar shows 'Aaron - Air Service Bot'. The main area displays nine product tiles: Account Kit, Facebook Login, Audience Network, Analytics, Messenger, Webhooks, Instant Games, Marketing API, and App Center. The 'Messenger' tile is highlighted with a red box.

Select Messenger option and click on **Set Up** to customize the way, to interact with people on Messenger through your app.

The screenshot shows the Facebook Developers website at <https://developers.facebook.com/apps/337146167016985/add/>. The 'Messenger' tile is selected, and its 'Set Up' button is highlighted with a red box. The rest of the interface is identical to the first screenshot.

Once you click on the Set Up, you will get the Messenger settings. In left side panel under Product, go to Messenger tile and select **Settings**.

APP ID: 337146167016985 Status: In Development

Messenger Platform

Welcome to the Messenger Platform!

The Messenger Platform offers a rich set of APIs, web plug-ins, and a complete webview that give you everything you need to build awesome experiences. Reach people across every device and platform they use, support multiple communication channels, create hybrid experiences that incorporate automated and live chat, and more, all in Messenger.

To ensure quality and prevent abuse of the Platform, all Messenger apps must be submitted for review before they can interact with everyone on Messenger. During development you will be able to send messages to anyone that has been granted either the Administrator, Developer or Tester role for your app.

To get started, check out our developer documentation for complete details on what the Platform has to offer, and be sure to read our Platform Policies for information on allowed usage and terms of use.

Increase Traffic to Messenger

Create ads to help more people discover your experience in Messenger. [Learn more](#)

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!

Scroll down to **Token Generation**, under **Page** choose the Facebook page that you want to use for your app.

Now, you will get a **Page Access Token** for your respective page. Copy that **Page Access Token**.

APP ID: 272814440092570 Status: In Development

Token Generation

Page token is required to start using the APIs. This page token will have all messenger permissions even if your app is not approved to use them yet, though in this case you will be able to message only app admins. You can also generate page tokens for the pages you don't own using Facebook Login.

Page	Page Access Token
Aaron	EAAD4H4ZCKk5oBAJlnzLwIA4WPdECqQDZA1wSSFTaryZCnMfudo2Luz9bz34lizAdTbYZC4Je0Jrw4yvS7gnFOG6dKpAP156Gp

Webhooks

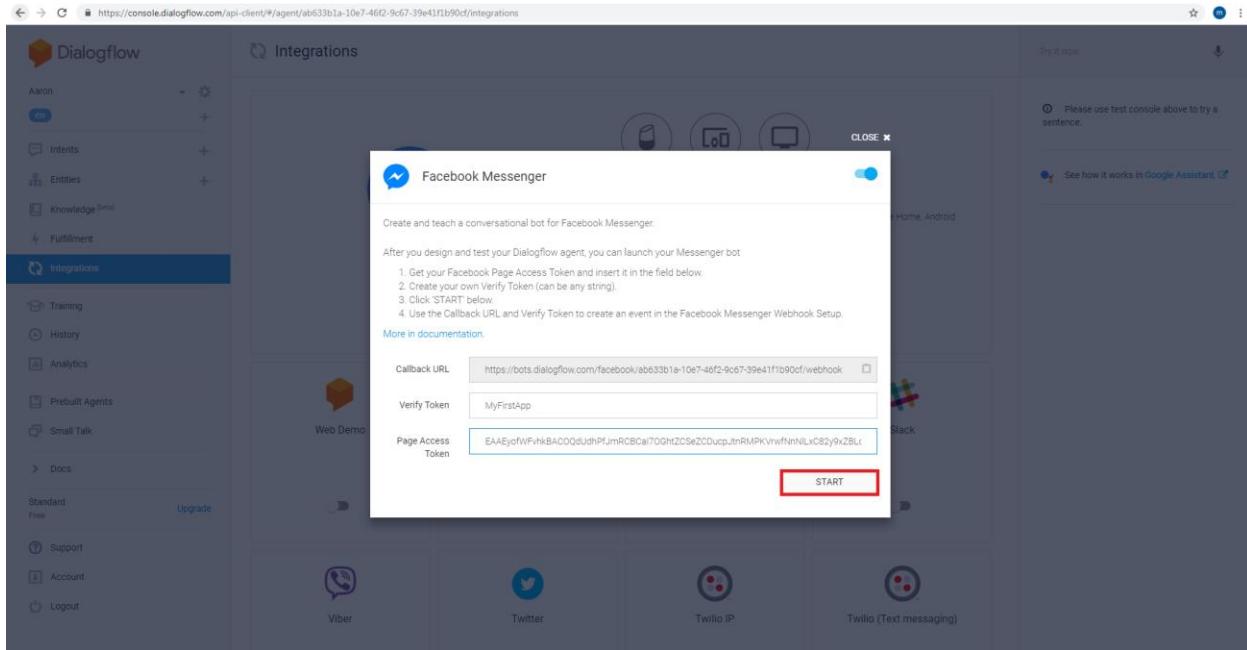
To receive messages and other events sent by Messenger users, the app should enable webhooks integration. [Setup Webhooks](#)

Built-In NLP

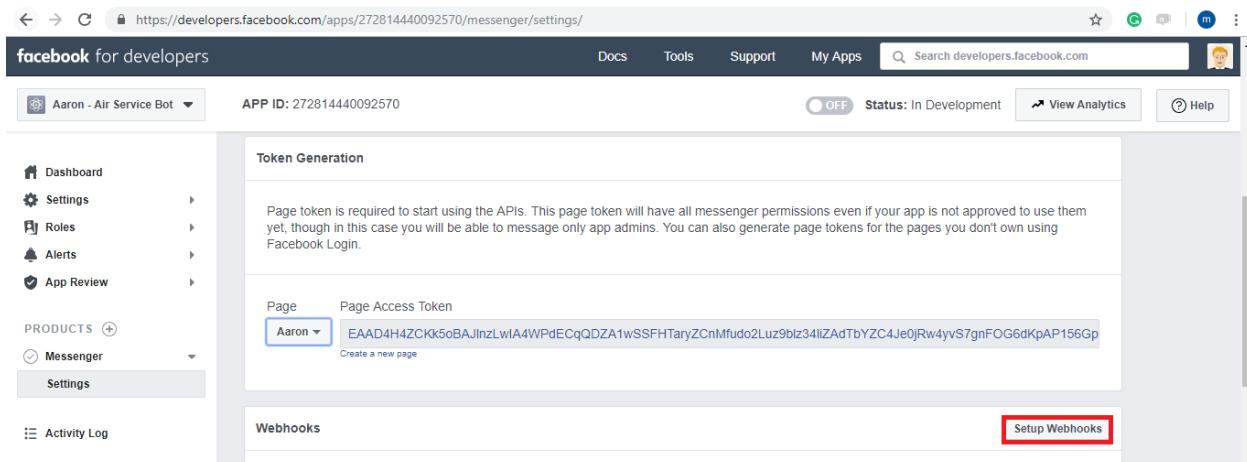
Built-in NLP (Natural Language Processing) parses the messages your Page receives to help detect meaning and extract information about dates, time and more. To automatically integrate built-in NLP start by choosing a Page. [Learn more](#)

Step #6 | Integration with Facebook Messenger and Dialogflow

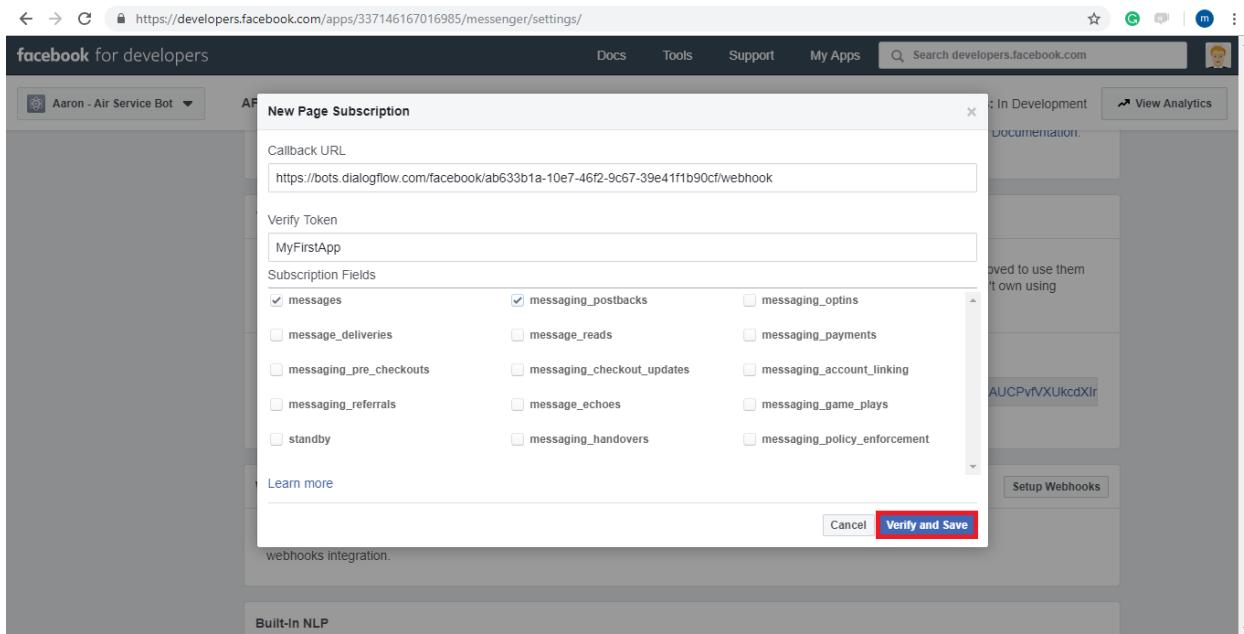
Navigate to Dialogflow Facebook Integrations and paste it over the **Page Access Token** Section. Provide your own **Verify Token** and click on **Start** button.



After clicking on start button in Dialogflow Facebook Integration, navigate to the app that you created in your Facebook Developer Page. Go to the Messenger settings and scroll down to the Webhooks section. Click on **Setup Webhooks**.



You will be navigated to **New Page Subscription** window, paste the copied callback URL in the specified field. In the **Verify Token** field, specify the same Facebook verify token that you provided in Dialogflow. Under **Subscription Fields**, select **messages** and **messaging_postbacks**. Now, click on **Verify and save**.



After the verification finishes, go back to the **Webhooks** section in the Messenger settings and click on Select a Page. Select the same page you selected during token generation, and then click **Subscribe**.

facebook for developers

APP ID: 272814440092570

OFF Status: In Development View Analytics Help

Webhooks

To receive messages and other events sent by Messenger users, the app should enable webhooks integration.

Selected events: **messages, messaging_postbacks** ✓ Complete

Select a page to subscribe your webhook to the page events **Aaron** Subscribe

The app is not subscribed to any pages

Built-In NLP

Built-in NLP (Natural Language Processing) parses the messages your Page receives to help detect meaning and extract information about dates, time and more. To automatically integrate built-in NLP, start by choosing a Page. [Learn more](#).

Subscribe a Page first under the "Webhooks" section to use Built-In NLP

Now, you will get the **Subscribed pages: <Name of the page you created>** in the Webhooks section as below,

facebook for developers

APP ID: 272814440092570

OFF Status: In Development View Analytics Help

Webhooks

To receive messages and other events sent by Messenger users, the app should enable webhooks integration.

Selected events: **messages, messaging_postbacks** ✓ Complete

Select a page to subscribe your webhook to the page events **Subscribed pages: Aaron** Unsubscribe

Built-In NLP

Built-in NLP (Natural Language Processing) parses the messages your Page receives to help detect meaning and extract information about dates, time and more. To automatically integrate built-in NLP, start by choosing a Page. [Learn more](#).

Select a Page

Step #7 | Integration with Node JS and Google's Dialogflow

Once you are done with the integration of Dialogflow with Facebook Messenger, navigate to the workspace folder where the code exists, and open command prompt.

Run **node app.js** to run your application.

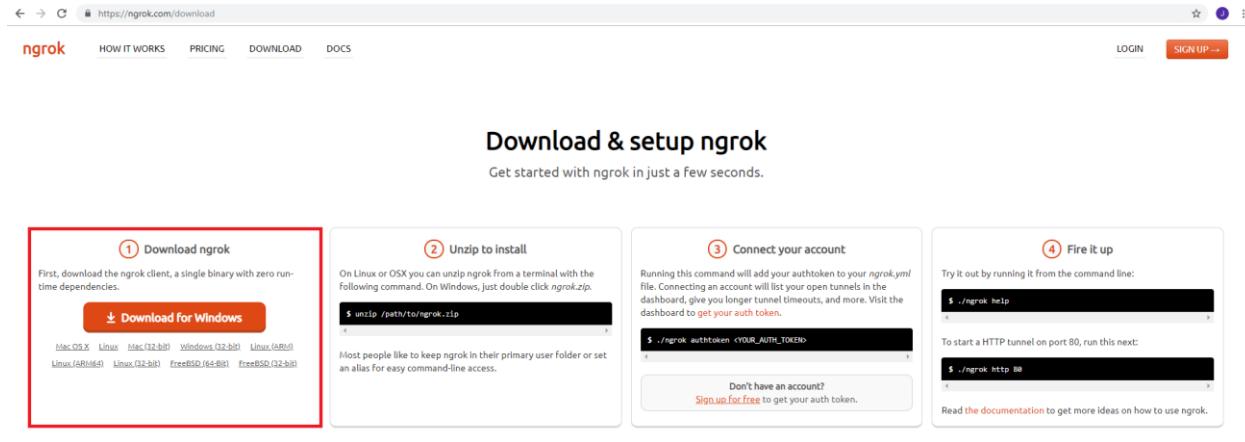


```
C:\Windows\System32\cmd.exe - node app.js
C:\Users\avennela\Desktop\DS'18 node app.js
running on 8000
```

Application is **running on 8000** port. Remember this for later use.

Open the browser and download ngrok by using the below link, <https://ngrok.com/download> (ngrok helps your app to get secure URL to your localhost server).

You need to install the ngrok based on the system requirement as provided below,



Download & setup ngrok
Get started with ngrok in just a few seconds.

- 1 Download ngrok**
First, download the ngrok client, a single binary with zero runtime dependencies.
[Download for Windows](#)
Mac OS X Linux Mac (32-bit) Windows (32-bit) Linux (ARM)
Linux (ARM64) Linux (32-bit) FreeBSD (64-Bit) FreeBSD (32-Bit)
- 2 Unzip to install**
On Linux or OSX you can unzip ngrok from a terminal with the following command. On Windows, just double click `ngrok.zip`.
`$ unzip /path/to/ngrok.zip`
Most people like to keep ngrok in their primary user folder or set an alias for easy command-line access.
- 3 Connect your account**
Running this command will add your auth token to your `ngrok.yml` file. Connecting an account will list your open tunnels in the dashboard, give you longer tunnel timeouts, and more. Visit the dashboard to [get your auth token](#).
`$./ngrok authtoken <YOUR_AUTH_TOKEN>`
Don't have an account?
[Sign up for free](#) to get your auth token.
- 4 Fire it up**
Try it out by running it from the command line:
`$./ngrok http 8000`
To start a HTTP tunnel on port 80, run this next:
`$./ngrok http 80`
Read the [documentation](#) to get more ideas on how to use ngrok.

After downloading the ngrok, extract the downloaded folder and run the **ngrok.exe** file. Now, give the following command and click on the Enter button.

ngrok http <Your-Application-Port>

Example: `ngrok http 8000`

Once you run the application successfully with the above command, you will get the url as https://<random_code>.ngrok.io. Copy this for later use.

Note: You should not close this window until you stop running your Node JS application

```
C:\Users\avennela\Desktop\Desktop\ngrOk.exe - ngrOk http 8000
ngrok by @inconschreivable
Session Status          online
Session Expires         7 hours, 59 minutes
Update                 update available (version 2.2.8, Ctrl-U to update)
Version                2.2.3
Region                 United States (us)
Web Interface          http://127.0.0.1:4040
Forwarding             http://2f558d60.ngrok.io -> localhost:8000
Forwarding             https://2f558d60.ngrok.io -> localhost:8000
Connections            ttl     opn      rt1     rt5     p50     p90
                        0       0       0.00   0.00   0.00   0.00
(Ctrl+C to quit)
```

Now, navigate to the Dialogflow Fulfillment section and enable the button which is showing as **DISABLED** initially as shown below,

The screenshot shows the Dialogflow console interface. On the left, there's a sidebar with navigation links like 'Integrations', 'Training', 'History', 'Analytics', 'Prebuilt Agents', 'Small Talk', 'Docs', 'Standard Free', and 'Upgrade'. The main area is titled 'Fulfillment' under 'Webhook'. It contains an 'Inline Editor (Powered by Cloud Functions for Firebase)' section with a code editor showing 'index.js' and 'package.json'. The code is a Node.js script for handling Dialogflow requests. A red box highlights the 'DISABLED' button at the top right of the fulfillment section. To the right of the fulfillment section, there are 'Try it now' and 'See how it works in Google Assistant' buttons.

If you enabled the fulfilment, need to provide the ngrok Url and it acts as a webhook which is an endpoint to your Node JS application.

Now, click on **Save** button.

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 password

HEADERS Enter value

DOMAINS Disable webhook for all domains

Inline Editor (Powered by Cloud Functions for Firebase) DISABLED

Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. Docs

```
index.js package.json
1 // This file is generated by Dialogflow's Fulfillment module.
2 // It's for handling fulfillment requests that don't fit into intents.
3 // See https://dialogflow.com/docs/fulfillment#fulfillment
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 DEBUG statements
10
11 exports.dialogflowFulfillment = functions.https.onRequest((request, response) => {
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(request.body)}`);
15
16   function welcome(agent) {
17     agent.add(`Welcome to my agent!`);
18   }
19
20   function fallback(agent) {
21     agent.add(`I don't understand.`);
22     agent.add(`I'm sorry, can you try again?`);
23   }
24
25   // If a message has both a text and a payload, Dialogflow will send them both.
26   // If a message has only a payload, Dialogflow will send it as a JSON object.
27   // Below is an example of how to handle both cases.
28   // Note: This section is from Dialogflow's Cloud Functions for Firebase authoring UI.
29   // agent.add(`This section is from Dialogflow's Cloud Functions for Firebase authoring UI.`);
30   // agent.add(new Card());
31
32   // If you want to add a card as a response
33   // agent.add(`https://www.google.com`);
34   // agent.add(`https://www.google.com`);
35
36   // If you want to add a suggestion
37   // agent.add(new Suggestion());
38
39   // If you want to add a simple text response
40   // agent.add(`Hello world!`);
41
42   // If you want to add a suggestion with a value
43   // agent.add(new Suggestion({value: 'Hello world!', suggestions: [Card.createText('Hello world!')]})
```

Now, click on **SAVE** button.

Fulfillment

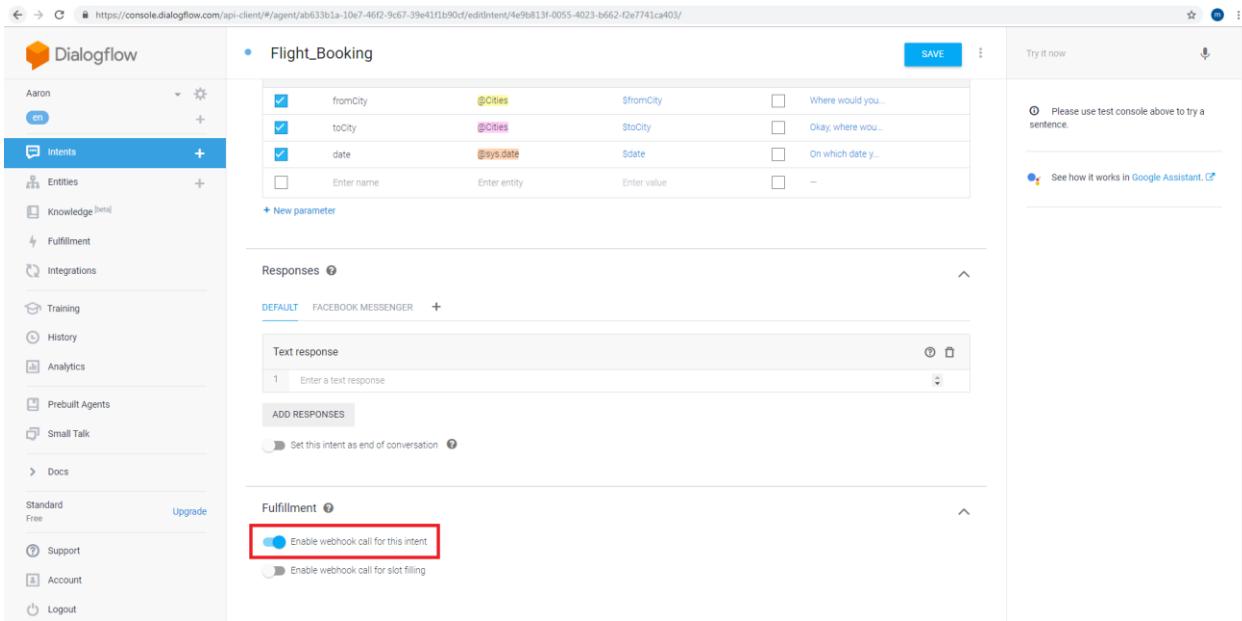
Inline Editor (Powered by Cloud Functions for Firebase) DISABLED

Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. Docs

```
index.js package.json
1 // This file is generated by Dialogflow's Fulfillment module.
2 // It's for handling fulfillment requests that don't fit into intents.
3 // See https://dialogflow.com/docs/fulfillment#fulfillment
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 DEBUG statements
10
11 exports.dialogflowFulfillment = functions.https.onRequest((request, response) => {
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(request.body)}`);
15
16   function welcome(agent) {
17     agent.add(`Welcome to my agent!`);
18   }
19
20   function fallback(agent) {
21     agent.add(`I don't understand.`);
22     agent.add(`I'm sorry, can you try again?`);
23   }
24
25   // If a message has both a text and a payload, Dialogflow will send them both.
26   // If a message has only a payload, Dialogflow will send it as a JSON object.
27   // Below is an example of how to handle both cases.
28   // Note: This section is from Dialogflow's Cloud Functions for Firebase authoring UI.
29   // agent.add(`This section is from Dialogflow's Cloud Functions for Firebase authoring UI.`);
30   // agent.add(new Card());
31
32   // If you want to add a card as a response
33   // agent.add(`https://www.google.com`);
34   // agent.add(`https://www.google.com`);
35
36   // If you want to add a suggestion
37   // agent.add(new Suggestion());
38
39   // If you want to add a simple text response
40   // agent.add(`Hello world!`);
41
42   // If you want to add a suggestion with a value
43   // agent.add(new Suggestion({value: 'Hello world!', suggestions: [Card.createText('Hello world!')]})
```

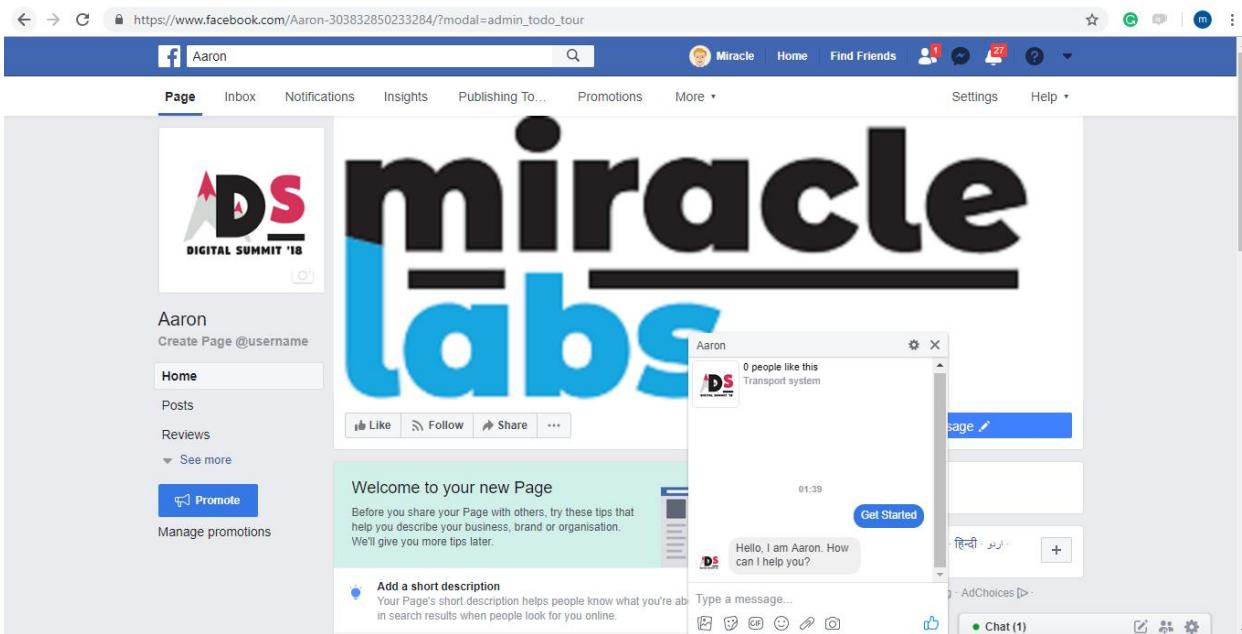
SAVE

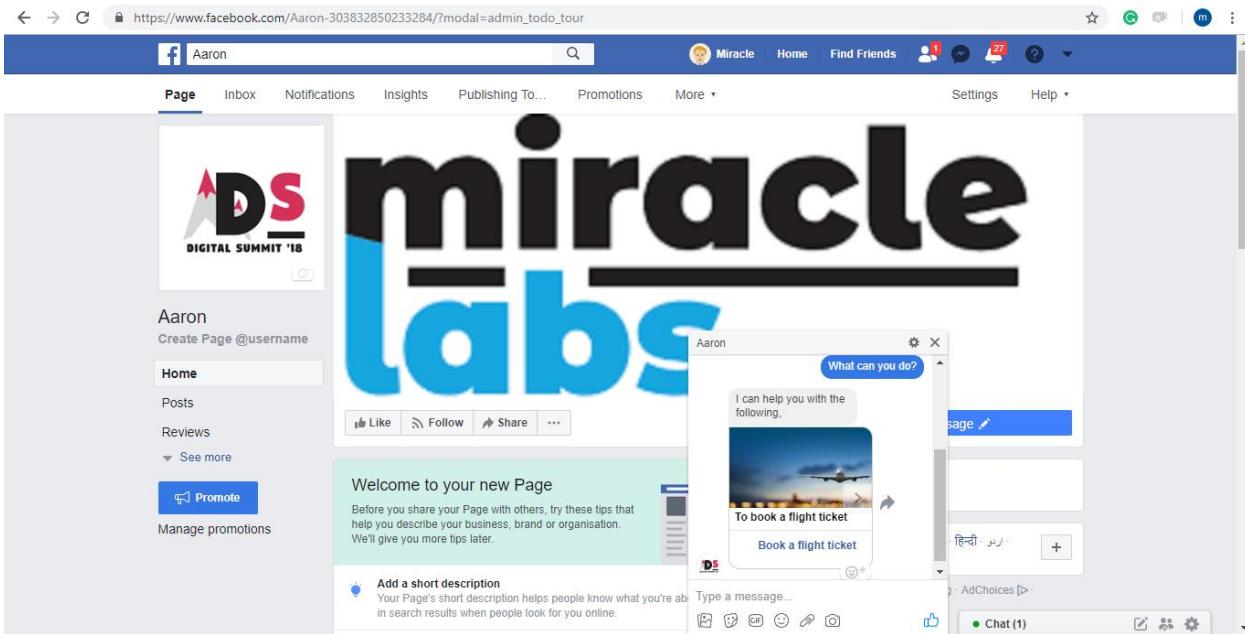
To get responses from your Node JS application to Dialogflow enable the fulfillment in all the required intent as shown below,



Step #8 | Testing Aaron Chat bot in Facebook Messenger

Open your Facebook Messenger, search for your Facebook bot (Or the Facebook page you subscribed) and talk to it!





Hurrah!! With this lab you were able to create your first chat bot with Google's Dialogflow using Facebook Messenger.

For any questions regarding the lab please feel free to reach out to innovation@miraclesoft.com. We hope you enjoyed creating bots with us!