



Building a Facebook Messenger Bot with Dialogflow

DS '17 Hands-On Cognitive Lab

Anusha Vennela

Cognitive Research Associate
Miracle Software Systems, Inc.

Facebook Messenger Bot with 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 Chat bot 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 and Firebase
- Access for Facebook and Facebook for Developers
- Facebook account to create a page for Chat Bot integration

Technology Involved

- Dialogflow
- Firebase Functions

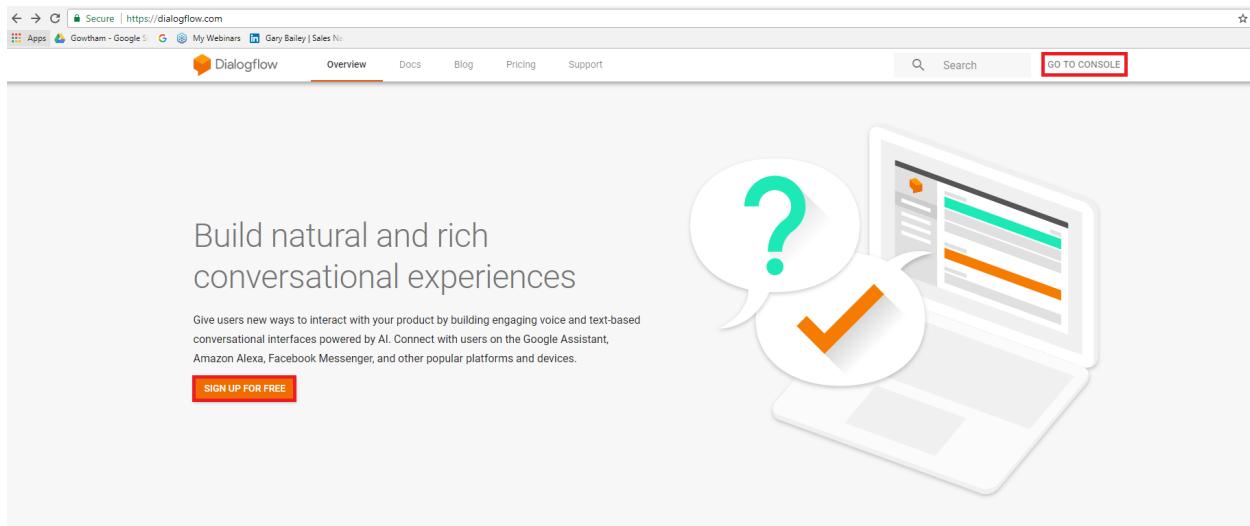
Labs Steps

So, let us get started with the bot!

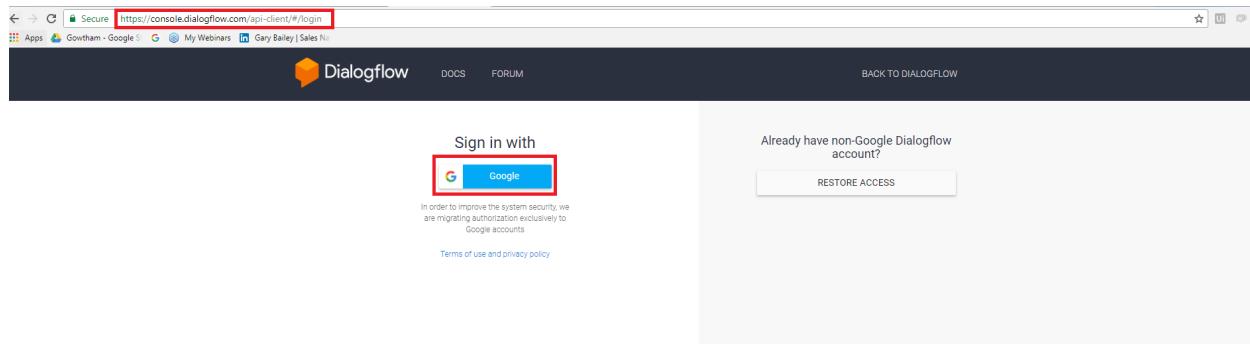
The following steps will outline how you can create a Tourist Bot and integrate it into Facebook Messenger. Users will be able to directly message your bot and perform operations such as "May I know the tourist places to visit in Goa."

1 | Create Dialog Model in Dialogflow

Now let's go and create an account for Dialogflow. Go to <https://dialogflow.com/> and click on **GO TO CONSOLE** or **SIGN UP FOR FREE**.



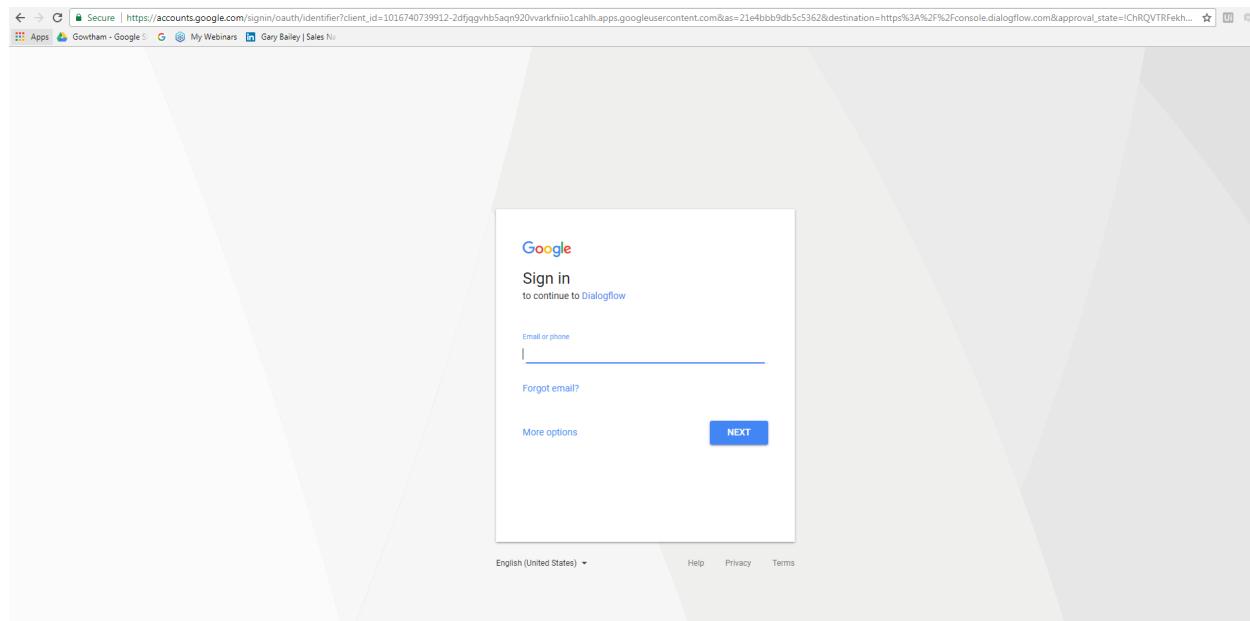
After clicking on any one of the button as above, it will be navigated to the console of Dialogflow for login. Now, click on **Google** button.



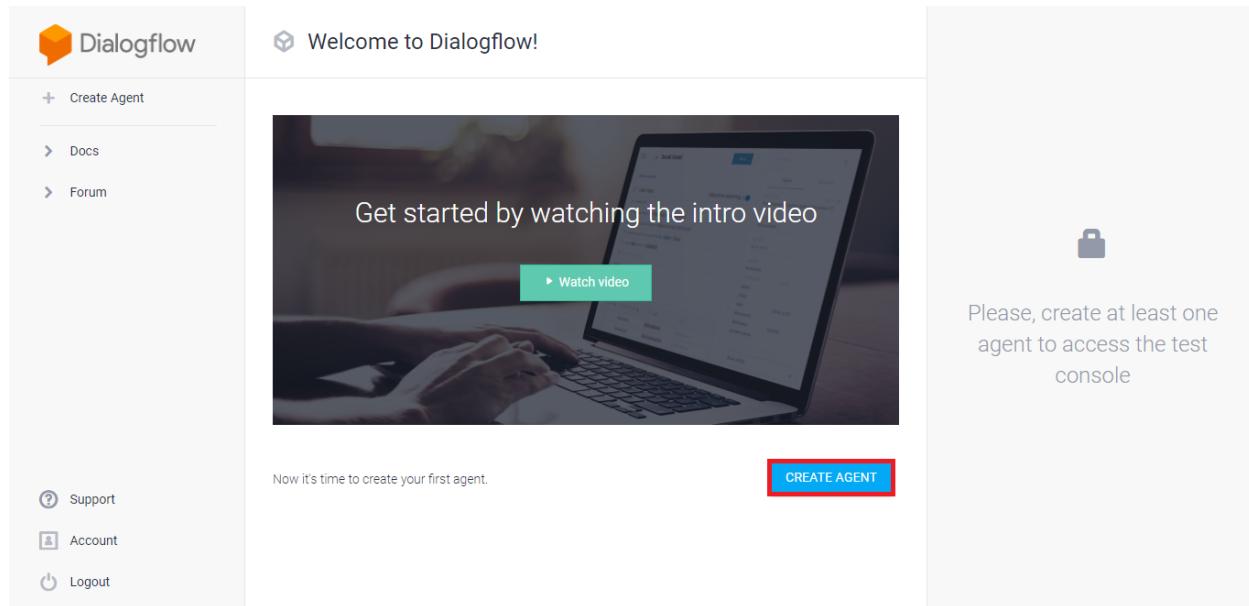
Dialogflow account can be created with Google account credentials. If you have a Google account then enter those credentials and authorize yourself.

If you don't have a Google account, create one at

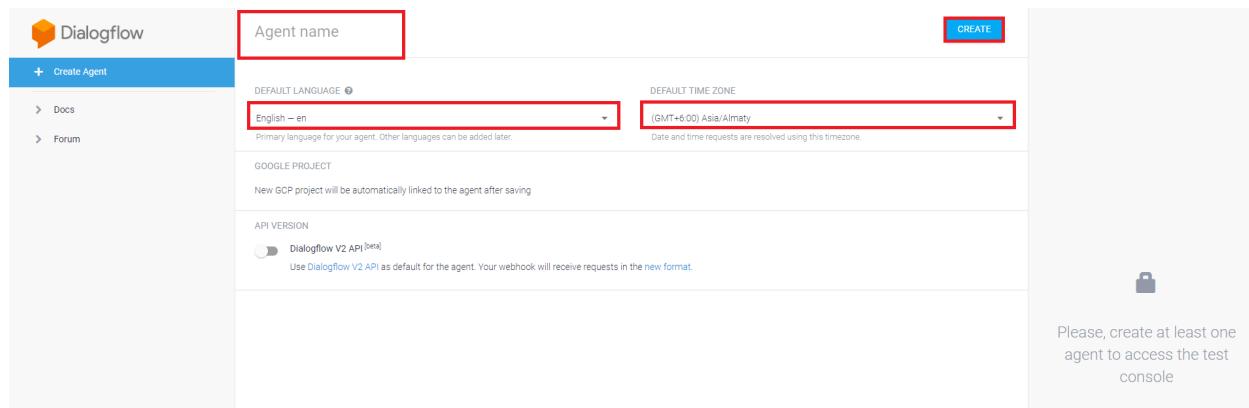
<https://accounts.google.com/SignUpWithoutGmail?hl=en/>



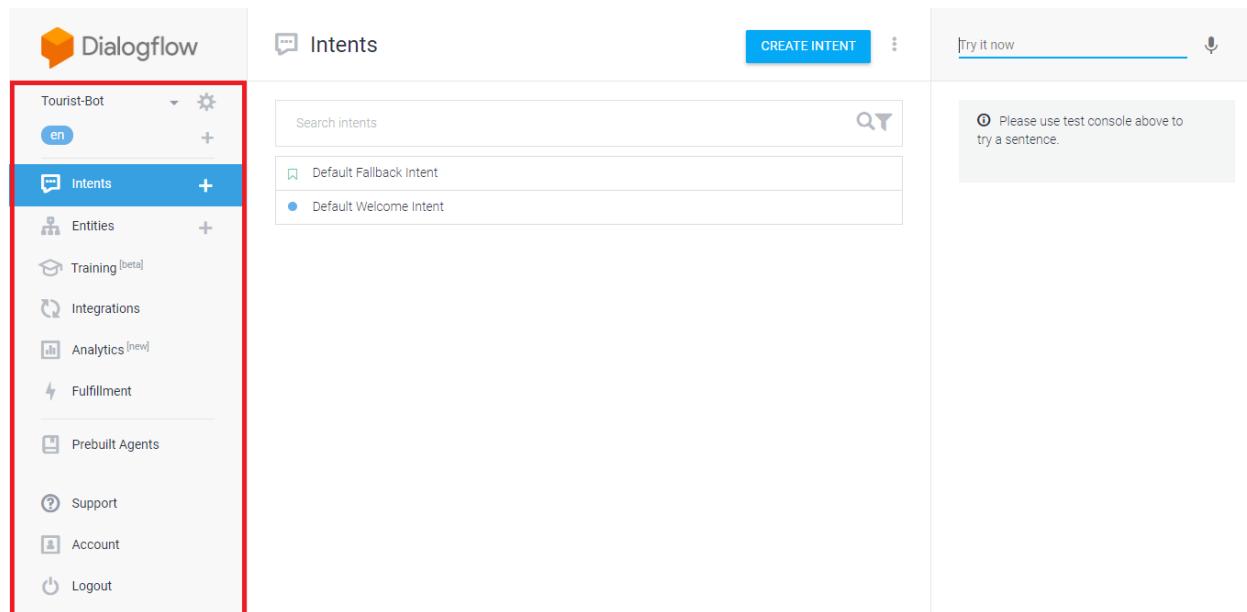
Now, you will get a Dialogflow home page where you have to create an agent/bot. Click on **CREATE AGENT**(if a pop up opens stating don't have correct permissions just Authorize).



Once you click on CREATE AGENT, then you need to provide an agent name (In our case it is Tourist-Bot), 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 and also a Default Welcome Intent shows up.

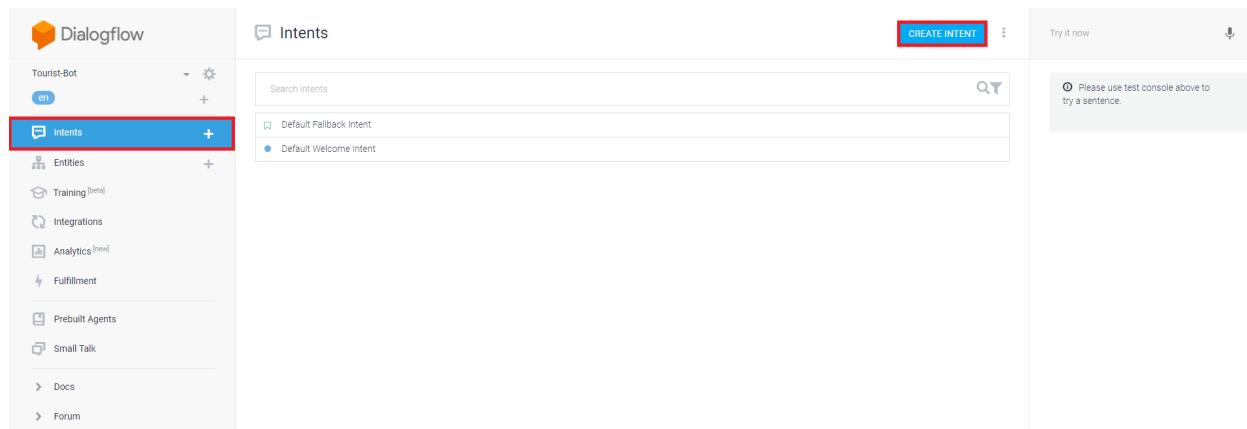


You can read about these in detail through [Dialogflow docs](#).

So 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.

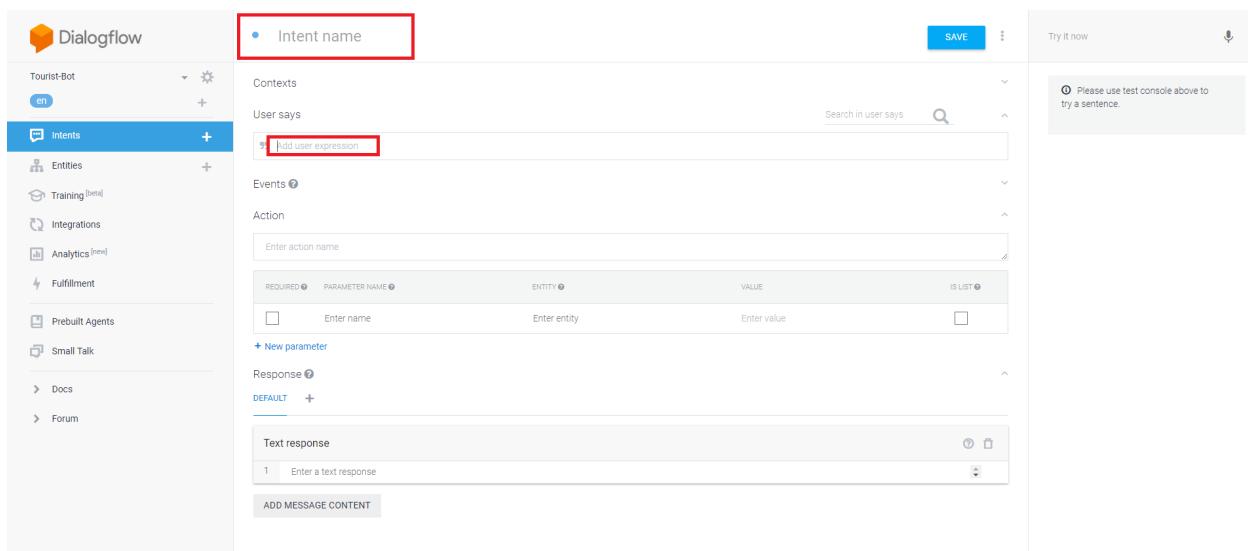
Creation of Intents

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

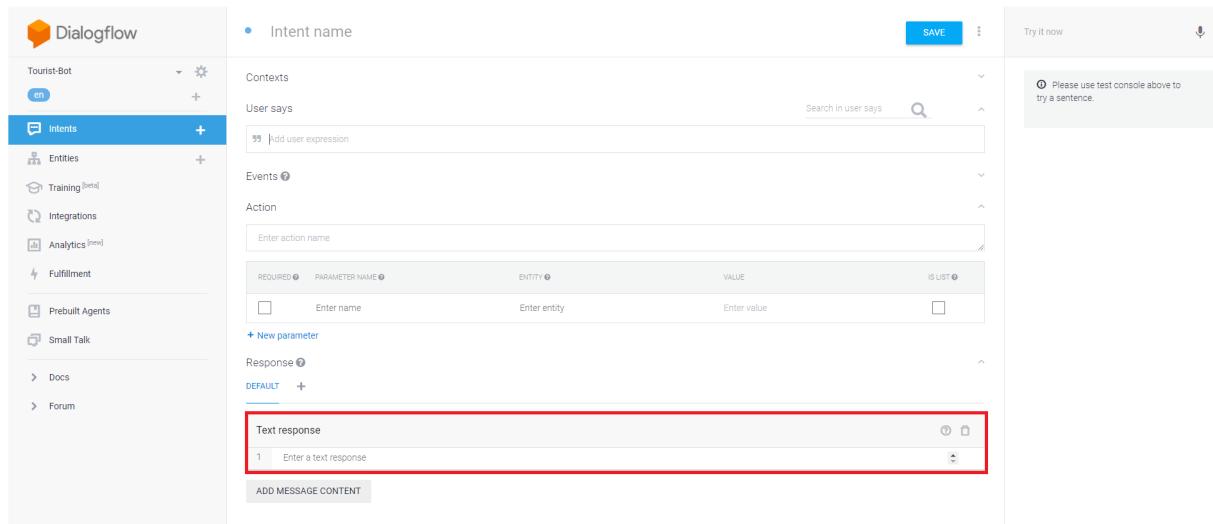


Now you need to provide the **Intent name** and **Add user expression** under User says section for creating the Intent

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

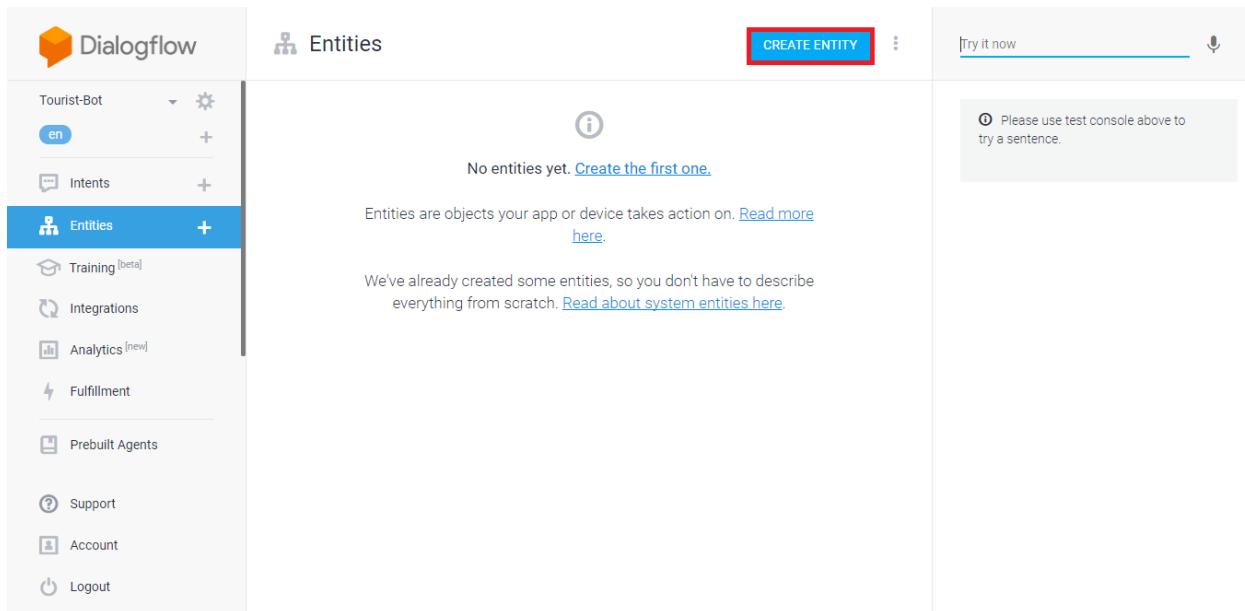


Now, provide the response in the **Text response** section which is used to respond back to the intent when it detects same kind of user examples. We can provide multiple responses for the same Intent. Once it is done, click on **SAVE** for saving the created intent.



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.



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 Dialogflow interface for creating a new entity. The left sidebar has 'Entities' selected. The main area is titled 'States' with a 'SAVE' button at the top right. A message says 'Please use test console above to try a sentence.' Below it, there's a text input field containing 'Andhra Pradesh' with a red box around it. Other suggestions like 'Andhra Pradesh', 'AP', and 'Andhra' are shown in dropdown boxes.

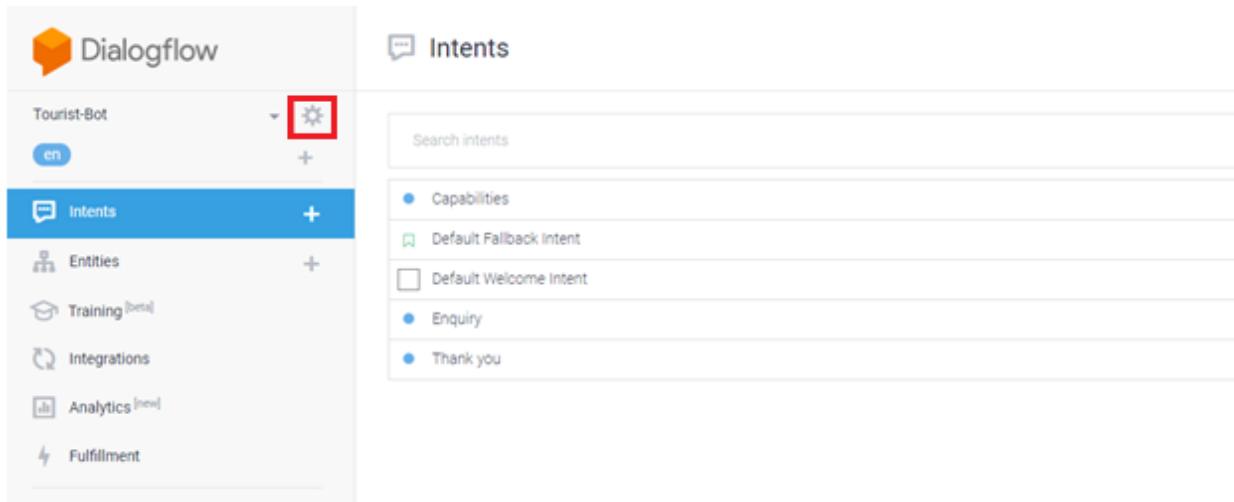
Then click on **SAVE** button on the top right to save the created entity.

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

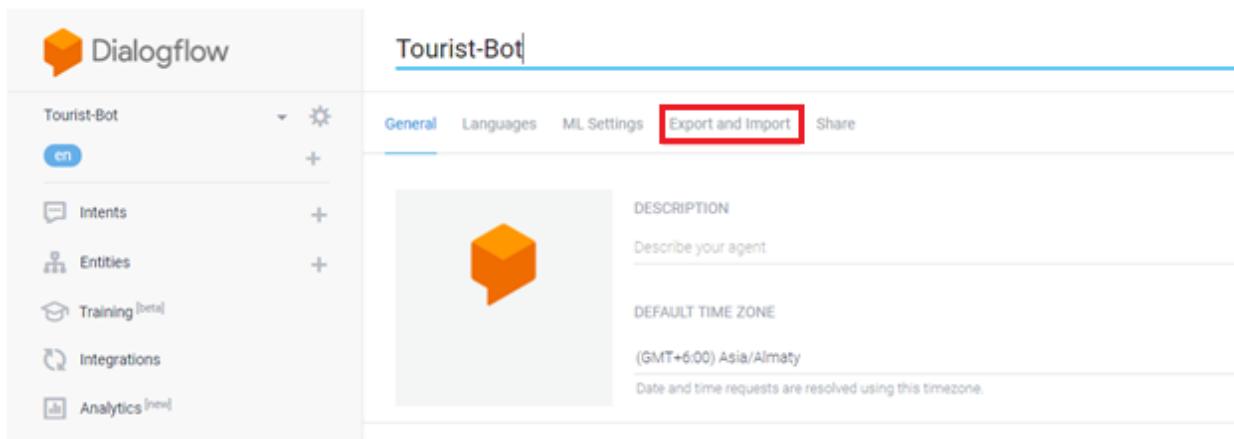
The screenshot shows the Dialogflow interface for configuring an intent. The left sidebar has 'Intents' selected. The main area shows a single intent named 'Default Welcome Intent' with a 'SAVE' button. The 'Events' section contains 'Hello' and 'Hi'. The 'Action' section shows 'input.welcome'. The 'Response' section shows a 'DEFAULT' response of 'Good day!'. The right side features a 'Try it now' test console with a red box around it, showing a user message 'Hi' and a bot response 'Good day!'. Below the test console, sections for 'INTENT', 'ACTION', and 'SHOW JSON' are visible.

You can find the sample Agent in the github repository which is created with the name **Tourist-Bot.zip** and download that zip file in your system.

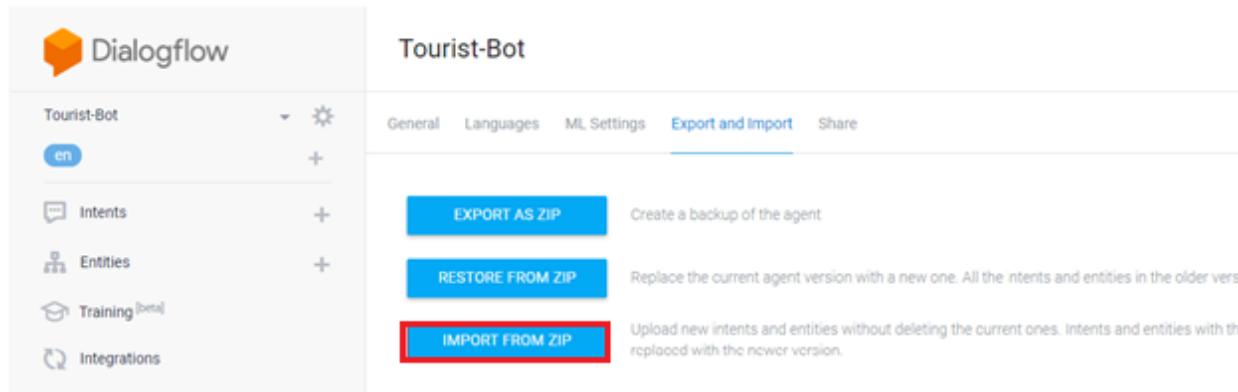
Now go to Dialogflow console and click on gear icon as shown below to import Intents and Entities.



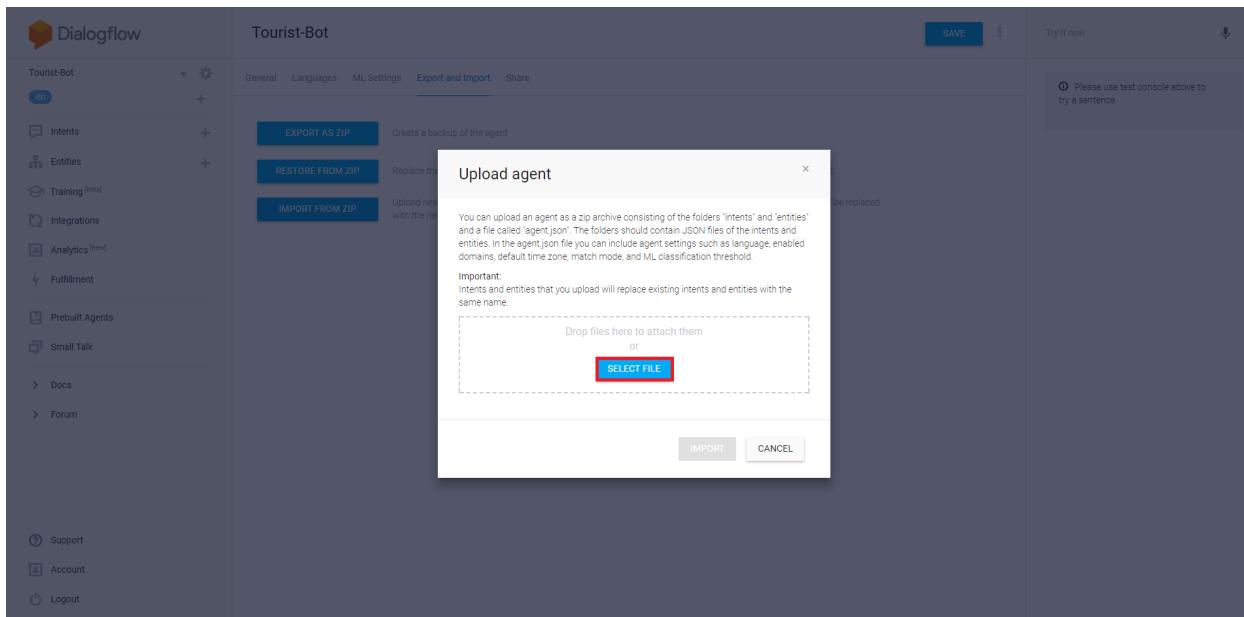
By default, you will be landed on General session page. Select **Export and Import** tab.



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



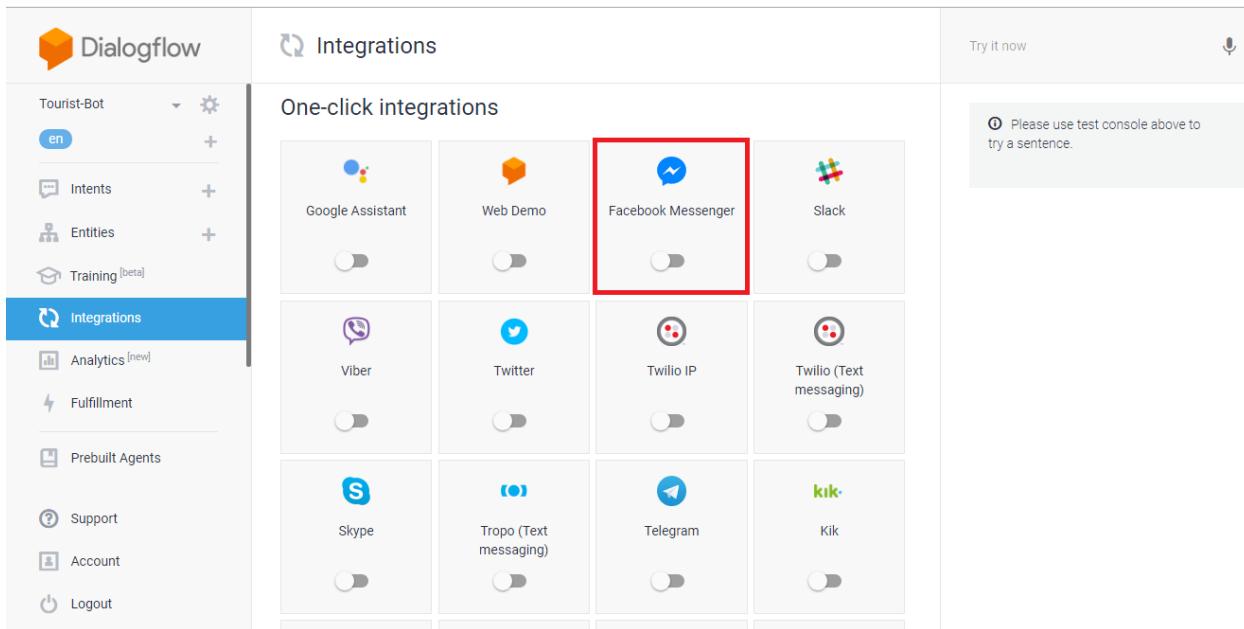
Now you will get a pop-up for asking you to **SELECT FILE**, you have to upload the zip file which you downloaded previously from github repository.
Once upload is done click on **IMPORT** button



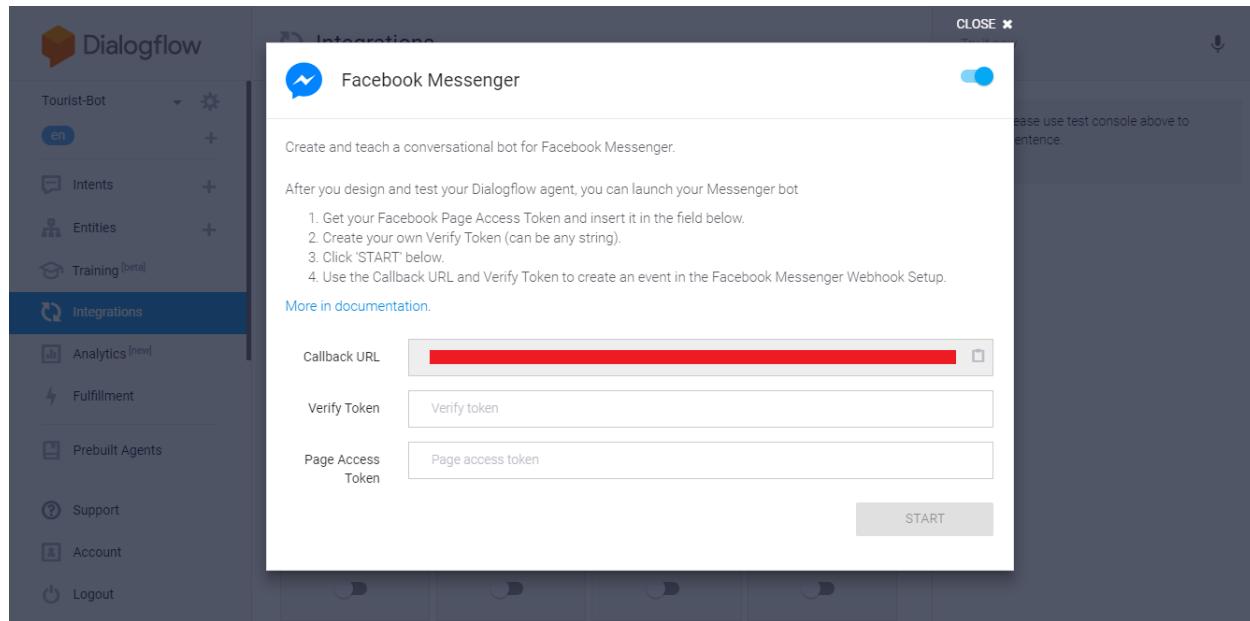
Once importing zip file is done you will find the agent of Tourist Bot which you imported.



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



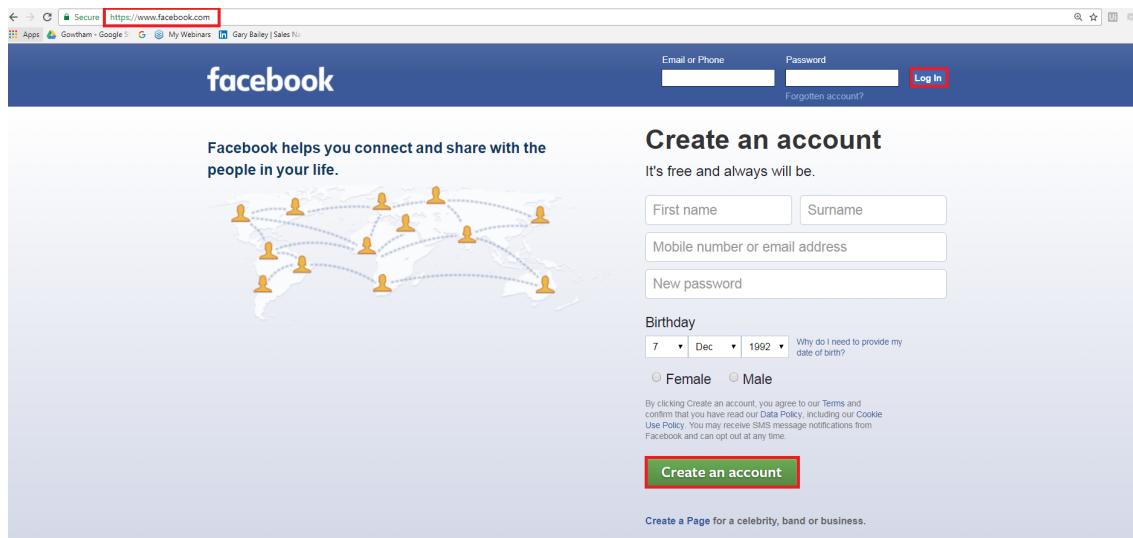
Once you click on Facebook Messenger integration, you will get a default Callback URL in a pop-up. Copy that Callback URL



#2 | Integrating agent with Facebook Messenger

The first step will be to create a Facebook account and **Log In** by providing Email or Phone and Password at,

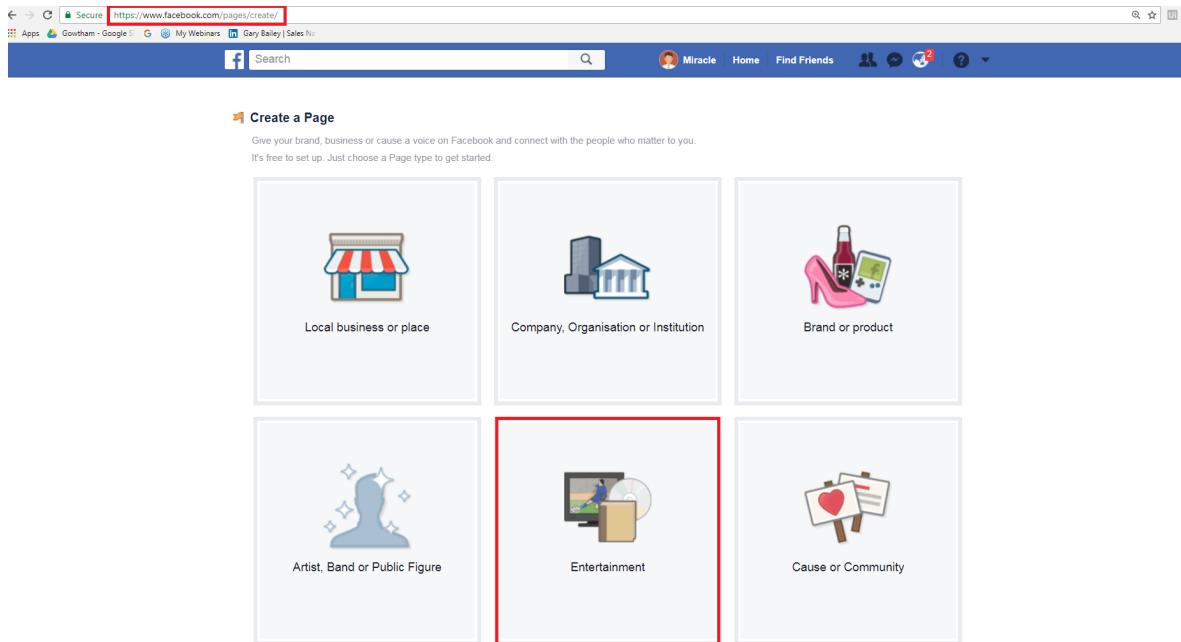
<https://www.facebook.com/>



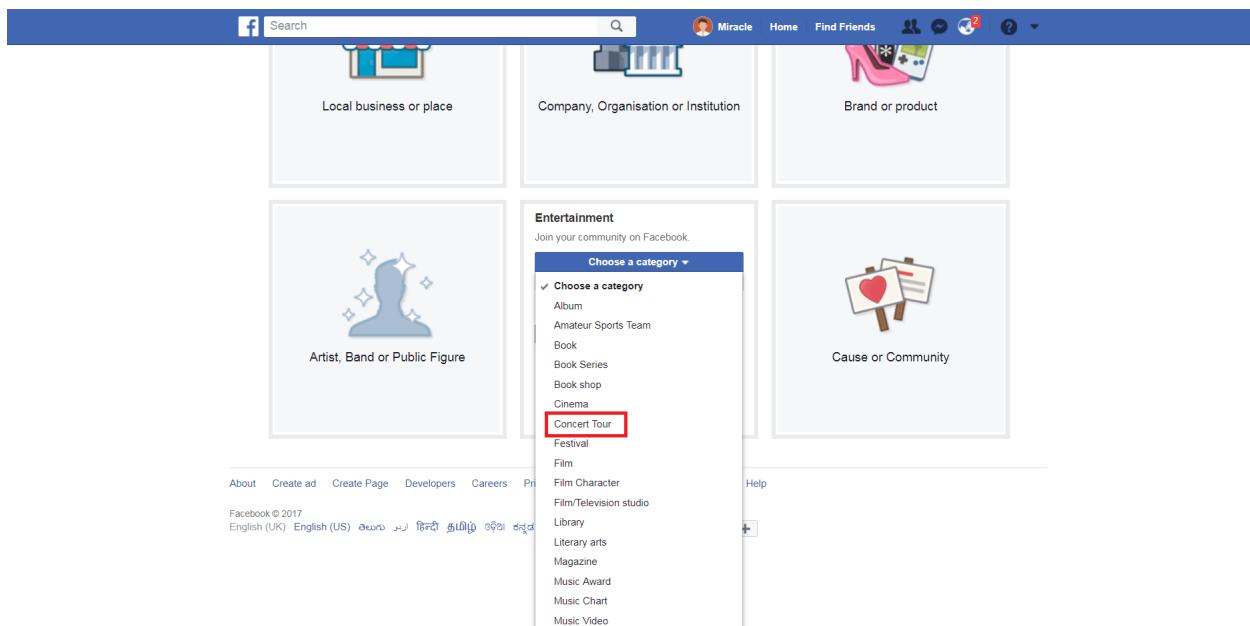
Once you login into Facebook, you need to create a new page by opening the below link in a new tab

<https://www.facebook.com/pages/create/>

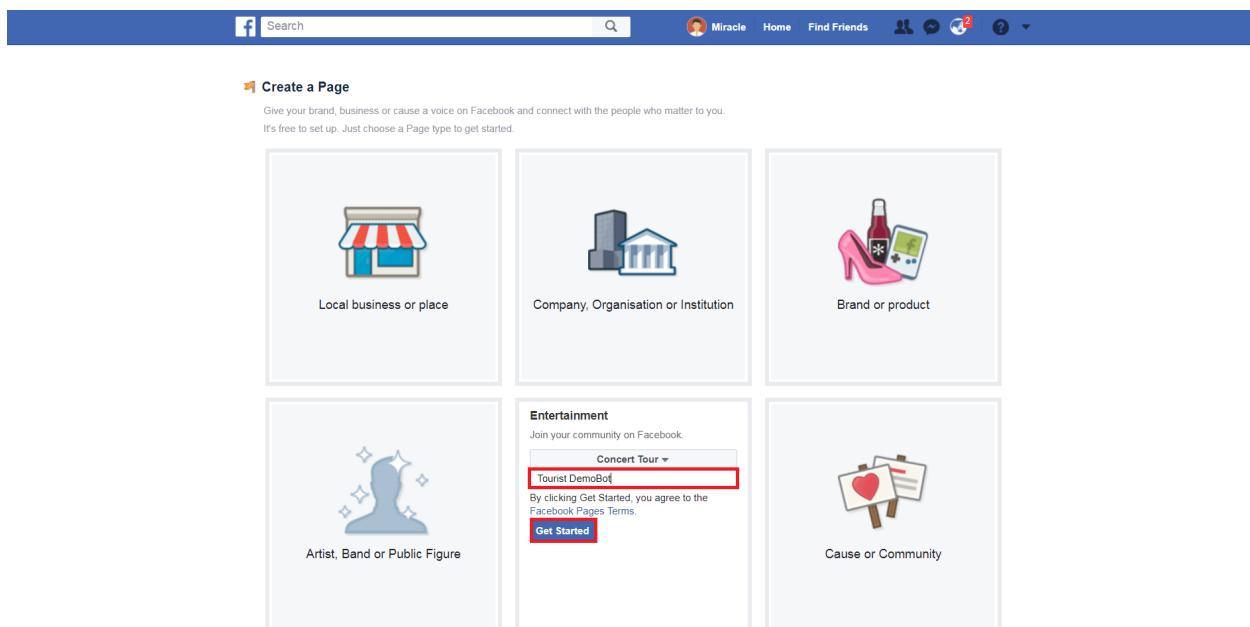
You need to select page type to get started. As Tourist Bot comes under entertainment category, select the **Entertainment** tile.



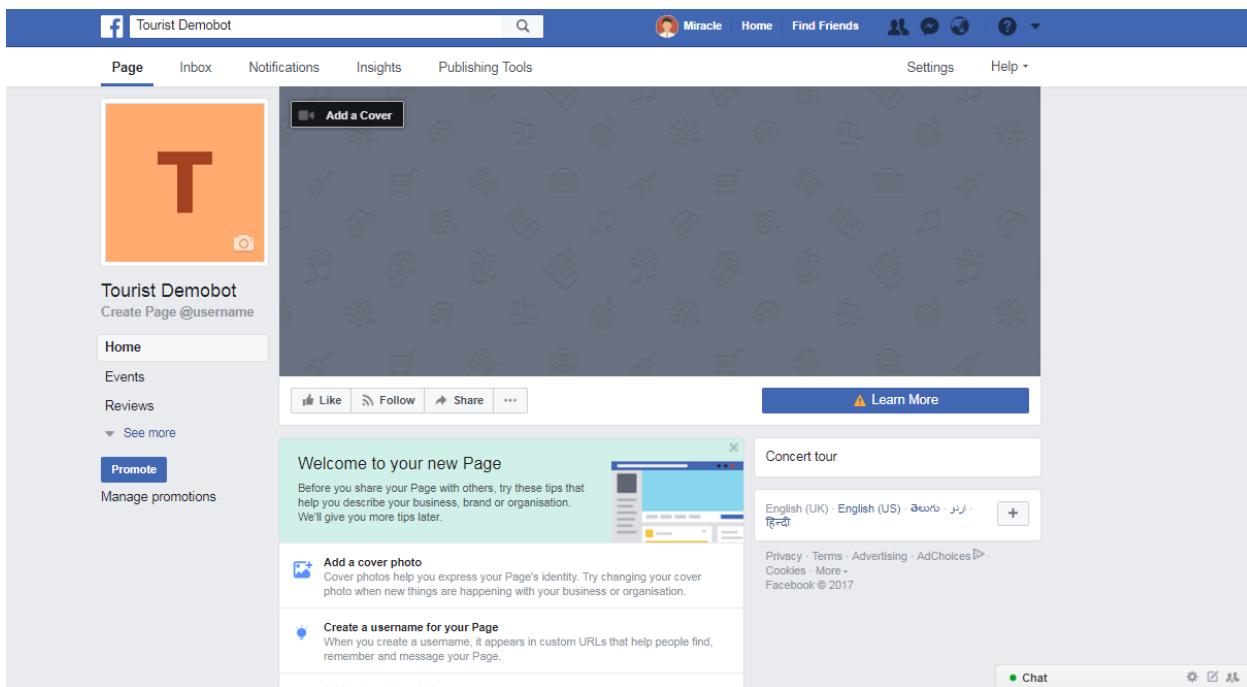
Now, from the dropdown list select **Concert Tour** under Choose a Category.



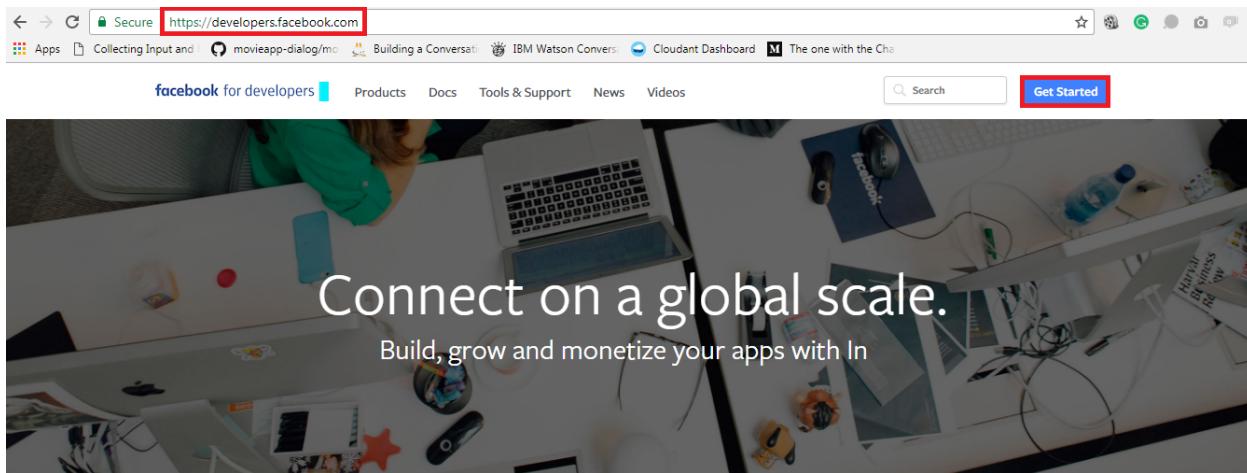
You need to provide a **Name** for creating a page. For Example: <Tourist DemoBot> and click on **Get Started**.



Your page will be created and looks like below,

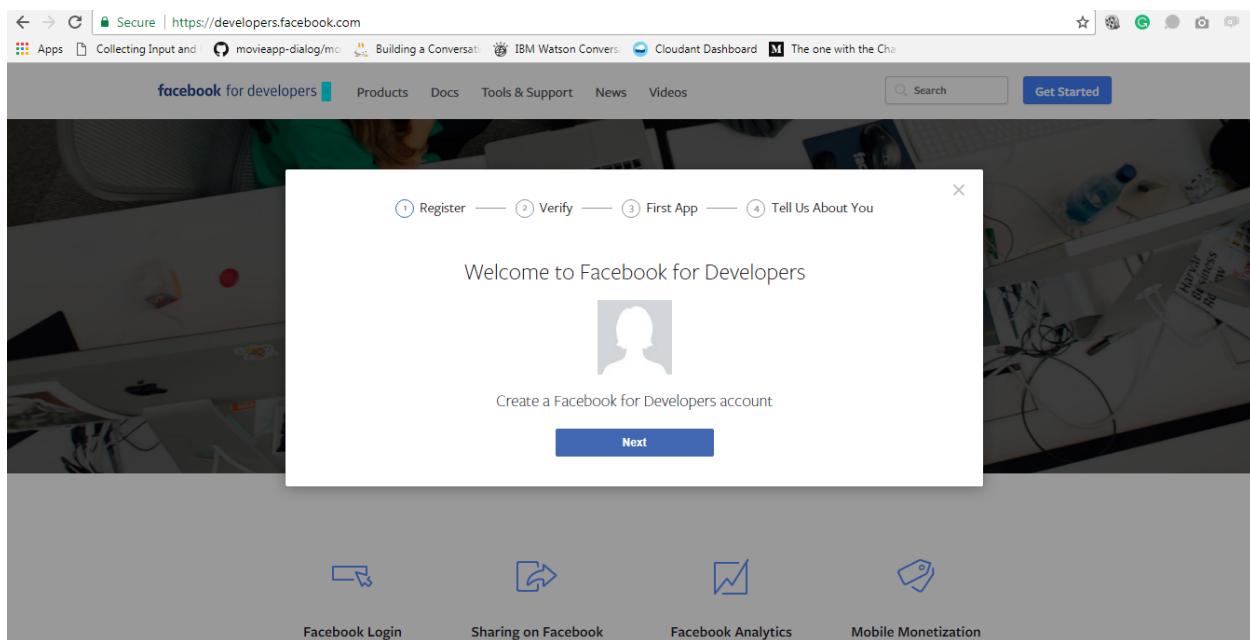


After your page is created, go to the <https://developers.facebook.com/> in a new tab and Click on **Get Started**



Complete all the 4 steps - **Register, Verify, First App** and **Tell Us About You**. For these steps, authenticate yourself by Email or Phone and provide <Your App Name> Example: “Miracle's Tourist Bot” as the app name.

Note : By default, it gives one app name and if you want to go with the app name which you are getting by default then follow the next steps. Orelse rename the app name.



Once it is done, you will land on the below page,

Secure | https://developers.facebook.com/apps/318398241974195/messenger/

APP ID: [REDACTED] View Analytics

Tools & Support Docs

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.

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!

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.

You need to have at least one page to generate page access token. Create a new page

Click on **+ Add Product** Choose Messenger and then click on **Set Up**.

Secure | https://developers.facebook.com/apps/318398241974195/messenger/

APP ID: [REDACTED] View Analytics

Tools & Support Docs

Messenger

Customize the way you interact with people on Messenger.

Webhooks

Subscribe to changes and receive updates in real time without calling the API.

Marketing API

Integrate Facebook Marketing API with your app.

App Center

Get your game discovered by being listed as a featured game on Facebook.

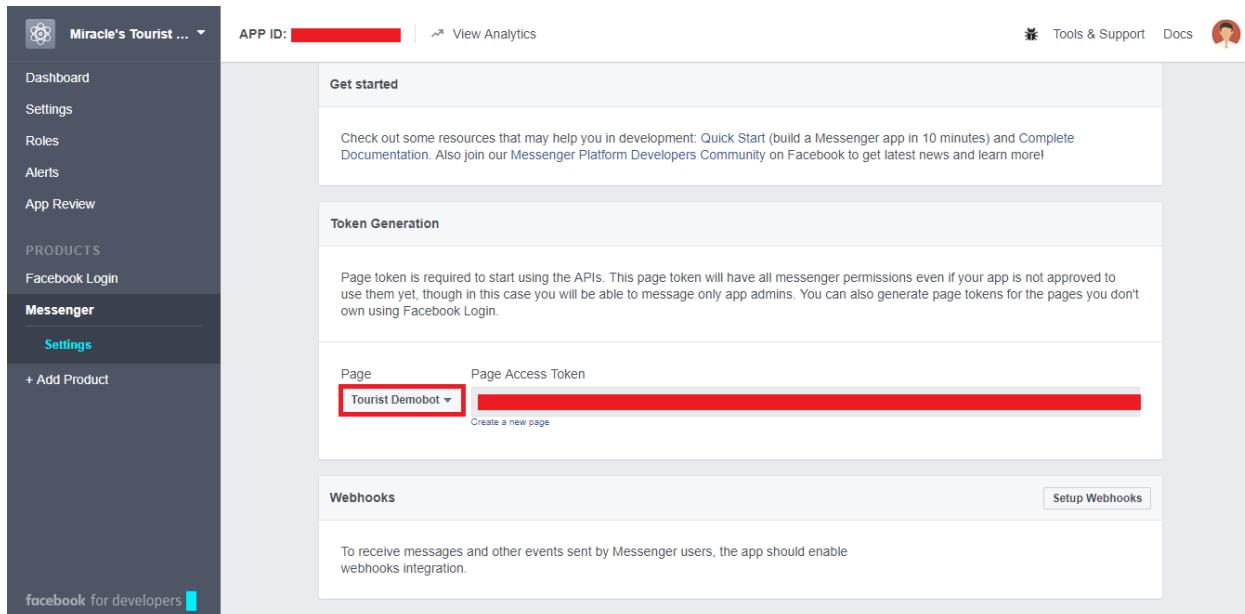
Web Payments

Accept in-app payments through Facebook's secure payment system.

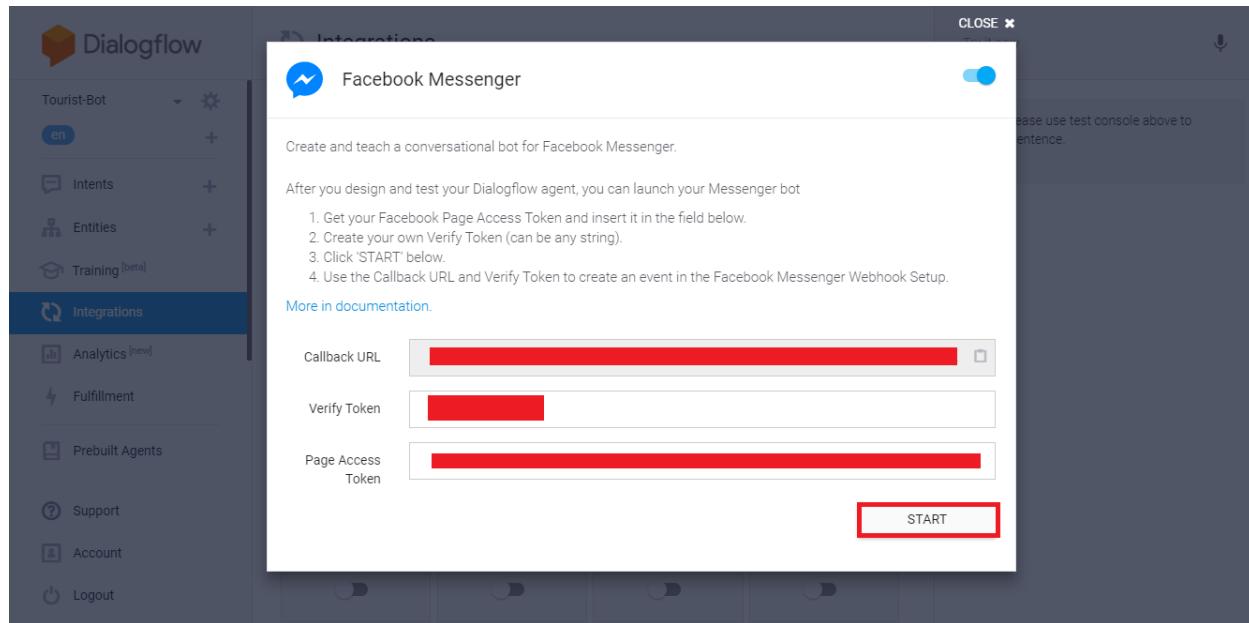
Instagram

Integrate your app with the Instagram API to let businesses use your app with their Instagram accounts.

After clicking on the Set Up, you will get the Messenger settings. Select the Facebook page you want the bot to be integrated from drop down list and you will get a Page Access token.

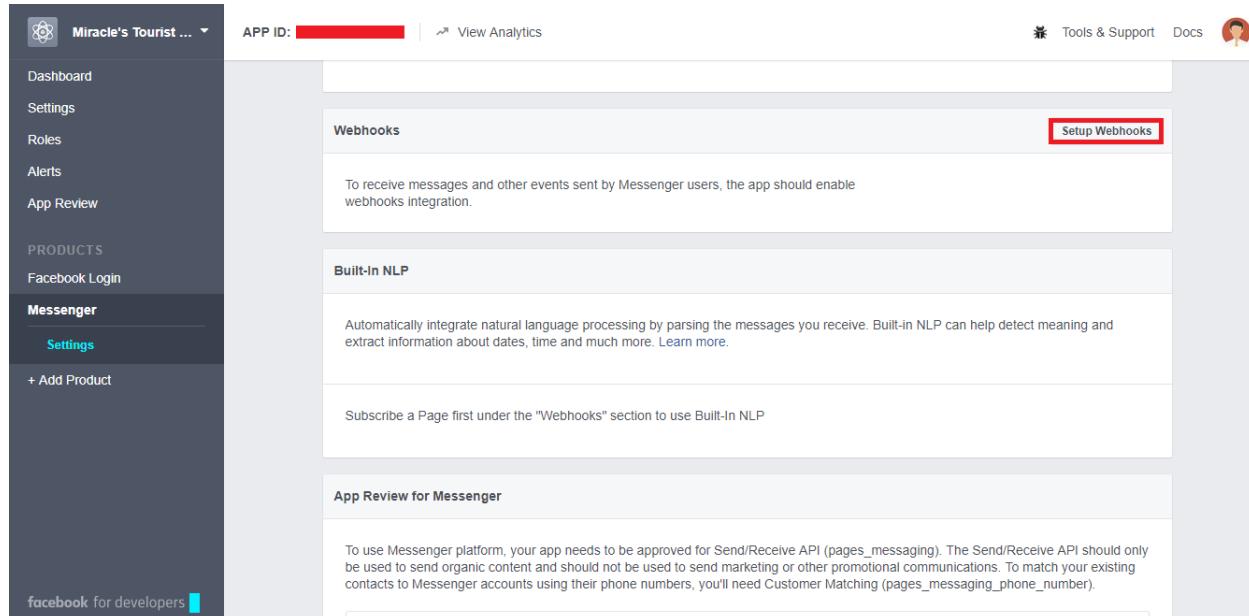


Copy the Page Access Token and switch to the Dialogflow page. Paste the Page Access Token in Facebook Messenger pop-up and give any random name in the Verify Token field Example: "Chat-Bot". Copy Callback URL and Verify Token for further use. Now click on **START** button.

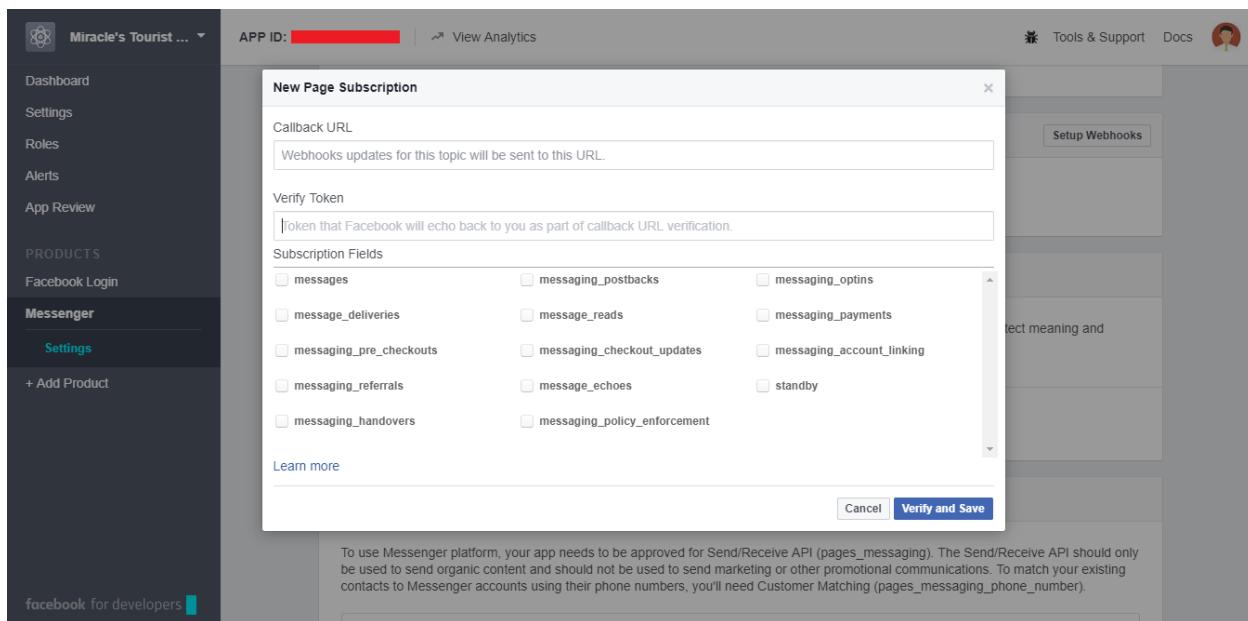


On successful start you will see a message in the bottom right of screen with the message "BOT started successfully".

Now, go back to Facebook Developer page and this time you need to add Webhook. So click on **Setup Webhooks** under Webhooks in Messenger Settings.



Now, you will get a pop-up like below, which is Webhooks landing page for new page subscription.



Copy the Callback URL and Verify Token from Dialogflow's Facebook Messenger tile and paste them in the respective fields and choose the options from Suggestion Fields and click on **Verify and Save**.

APP ID: [REDACTED] | View Analytics

New Page Subscription

Callback URL: [REDACTED]

Verify Token: [REDACTED]

Subscription Fields

- messages
- message_deliveries
- messaging_pre_checkouts
- messaging_referrals
- messaging_handovers
- messaging_postbacks
- message_reads
- messaging_checkout_updates
- message_echoes
- messaging_optins
- messaging_payments
- messaging_account_linking
- standby
- messaging_policy_enforcement

Learn more

Cancel Verify and Save

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

Once you are done with Webhook Setup, select Facebook page from dropdown list and then click on **Subscribe** button.

APP ID: [REDACTED] | View Analytics

Webhooks

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

Selected events: messages

Complete

Select a page to subscribe your webhook to the page events

The app is not subscribed to any pages

Built-In NLP

Automatically integrate natural language processing by parsing the messages you receive. Built-In NLP can help detect meaning and extract information about dates, time and much more. Learn more.

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

App Review for Messenger

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

The screenshot shows the left sidebar with 'Messenger' selected. In the main area, under 'Webhooks', it says 'To receive messages and other events sent by Messenger users, the app should enable webhooks integration.' Below this, it shows 'Selected events: messages' and 'Select a page to subscribe your webhook to the page events'. A dropdown menu shows 'Tourist Demobot' and a blue 'Unsubscribe' button. A red box highlights the 'Subscribed pages: Tourist Demobot' field.

#3 | Enable Fulfillment in Dialogflow using Inline Editor

Now go to Dialogflow Fulfillment section and enable Inline Editor. Here you can see the default Functions, lets write and manage backend code.

The screenshot shows the Dialogflow interface with 'Tourist-Bot' selected. The left sidebar has 'Fulfillment' selected. In the main area, the 'Fulfillment' tab is open, showing the 'Inline Editor' section. It says 'Build and manage fulfillment directly in Dialogflow via Cloud Functions for Firebase. Docs'. The 'index.js' file contains the following code:

```

1 // use strict;
2 const functions = require('firebase-functions'); // Cloud Functions for Firebase library
3 const DialogflowFulfillment = require('actions-on-google').DialogflowFulfillment; // Google Actions helper library
4 exports.dialogflowFulfillment = functions.https.onRequest((request, response) => {
5   console.log('Dialogflow Request headers: ' + JSON.stringify(request.headers));
6   console.log('Dialogflow Request body: ' + JSON.stringify(request.body));
7   if (request.body.result) {
8     processV1Request(request, response);
9   } else {
10     console.log('Invalid Request');
11     return response.status(400).end('Invalid Webhook Request (expecting v1 or v2 webhook request)');
12   }
13 });
14 /**
15 * Function to handle v1 webhook requests from Dialogflow
16 */
17 exports.processV1Request = (request, response) => {
18   let action = request.body.result.action; // https://dialogflow.com/docs/actions-and-parameters
19   let parameters = request.body.result.parameters; // https://dialogflow.com/docs/actions-and-parameters
20   let intentName = request.body.result.intent_name; // https://dialogflow.com/docs/intents
21   let requestSource = (request.body.originalRequest) ? request.body.originalRequest.source : undefined;
22   const googleAssistantRequest = 'google'; // Constant to identify Google Assistant requests
23   const dialogflowApp = 'dialogflow'; // Constant to identify Dialogflow App requests
24   // Create new DialogflowApp() if you want to use Dialogflow actions as well as a 'default' header
25   const actionHandlers = {
26     // If the default welcome intent has been matched, welcome the user (https://dialogflow.com/docs/events#default_welcome_intent)
27     'Default': () => {
28       // Use the Actions on Google lib to respond to Google requests; for other requests use JSON
29       if (request.body.result.parameters.State == "Andhra Pradesh") {
30         let responseToUser = '';
31         let responseToUser = {
32           // Uncomment to enable rich responses
33           // 'richResponse': { // Optional, uncomment to enable
34             // 'listItem': { // Optional, uncomment to enable
35               'text': 'Here are the best tourist places to visit in Andhra:\n' +
36               'Aruku Valley,\n' + 'Vizag,\n' + 'Srirajam Dam,\n' + 'Pepikondalu,\n' + 'Talakona Waterfalls.\n', // spoken response
37               'text': 'This is from Dialogflow's Cloud Functions for Firebase editor! :-)' // displayed response
38             },
39           'sendResponse(responseToUser);
40         }
41       }
42     }
43   }
44   // Last deployed on 12/06/2017 07:49
45   // Deploy
46 }

```

The right side of the screen shows a 'Try it now' button and a text input field with placeholder text: 'Please use test console above to try a sentence.'

Import the required modules

```
var express = require('express'); // requiring express module for routing and to to  
create REST APIs  
var app = express(); //Creating reference  
var bodyParser = require('body-parser'); // body-parser for post HTTP calls  
var port = process.env.PORT || 8000; //assigning dynamic or static port  
//using body-parser middleware  
app.use(bodyParser.urlencoded({  
    extended: true  
}));  
app.use(bodyParser.json());
```

Here is the HTTP post Webhook which will listen the request from Dialogflow

```
//post HTTP call for configuring Webhook in Dialogflow  
  
app.post('/test', function(req, res) {  
    if (req.body.result.action == "<action-name>") {  
  
        if (req.body.result.parameters.parameter_name == "Parameter_value") {  
            var message ="A Sample message from NodeJS to Dialogflow";  
  
        }  
    }  
})
```

Return response to Dialogflow

```
return res.json({  
    speech: message,  
    displayText: message,  
    source: 'webhook-echo-sample'  
});
```

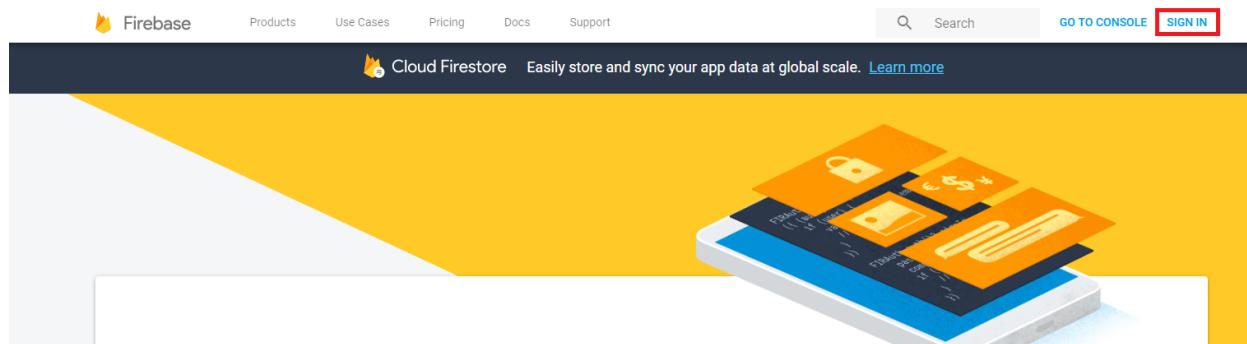
Server is listening at a particular port

```
app.listen(port, function() {  
  console.log('Tourist Bot is running on port:8000');  
});
```

Note : Backend code for this bot is available in Github repository. Copy the code from **app.js** file and paste it in Inline editor's Index.js file

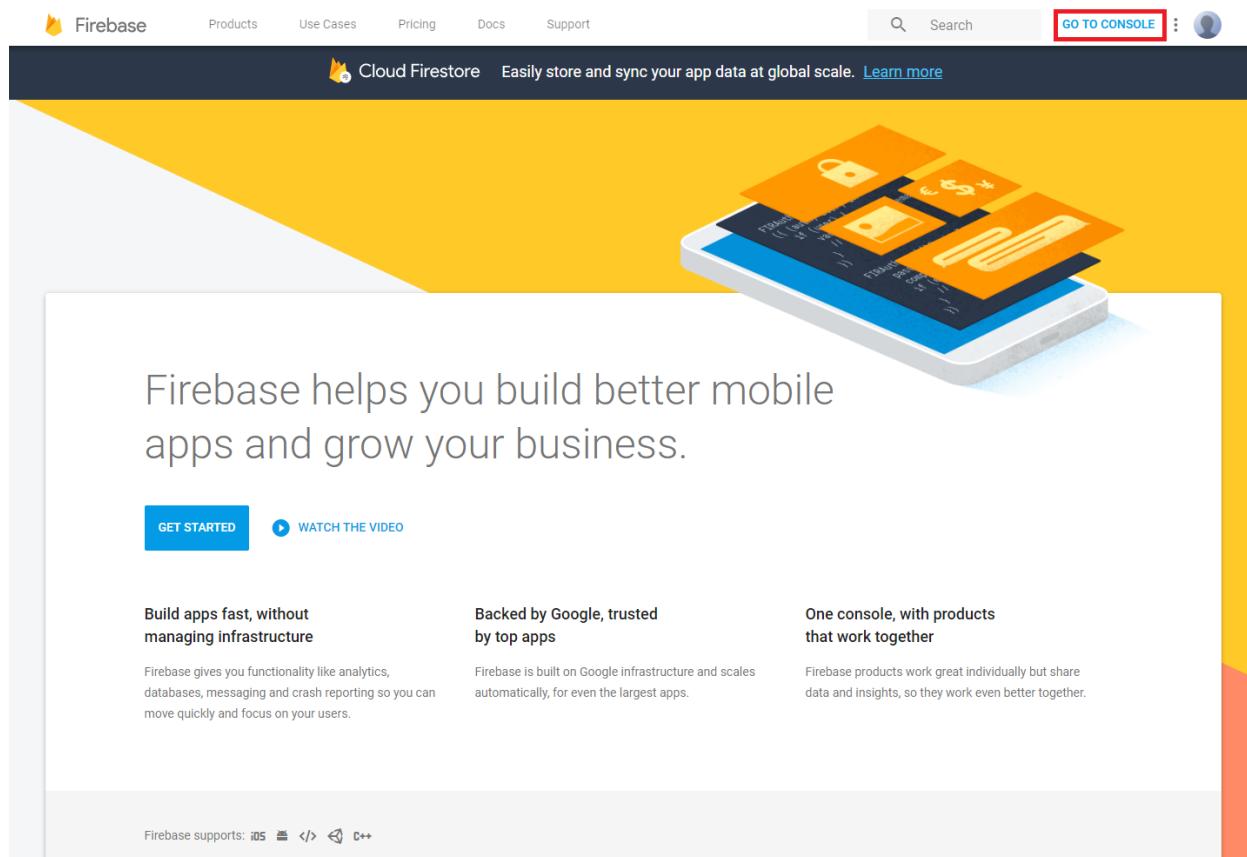
#4 | Deploy Functions in Firebase

Go to <https://firebase.google.com/> and click on **SIGN IN**. You have to provide credentials of Google account which you already used for Dialogflow.



The screenshot shows the official Firebase website. At the top, there is a navigation bar with links for Products, Use Cases, Pricing, Docs, and Support. On the right side of the nav bar are a search bar, a 'GO TO CONSOLE' button, and a 'SIGN IN' button (which is highlighted with a red box). Below the navigation, there is a banner for 'Cloud Firestore' with the subtext 'Easily store and sync your app data at global scale.' and a 'Learn more' link. The main content area features a large yellow background with a white smartphone icon displaying various app icons like a lock, a dollar sign, and a gear. The text 'Firebase helps you build better mobile apps and grow your business.' is centered above the phone. Below the phone are two calls-to-action: 'GET STARTED' and 'WATCH THE VIDEO'. There are three columns of text below these buttons: 'Build apps fast, without managing infrastructure' (with a note about Firebase supporting multiple platforms), 'Backed by Google, trusted by top apps' (with a note about its use in large-scale applications), and 'One console, with products that work together' (with a note about data sharing between products). At the bottom of the page, there is a note about supported platforms: 'Firebase supports: iOS, Android, JavaScript, Python, Java, .NET, C#, C++, and Go.'

Once you SIGN IN into Firebase then click on **GO TO CONSOLE** then it navigates to your projects dashboard in Firebase or else go through this link <https://console.firebaseio.google.com/> to get your projects dashboard in Firebase directly.

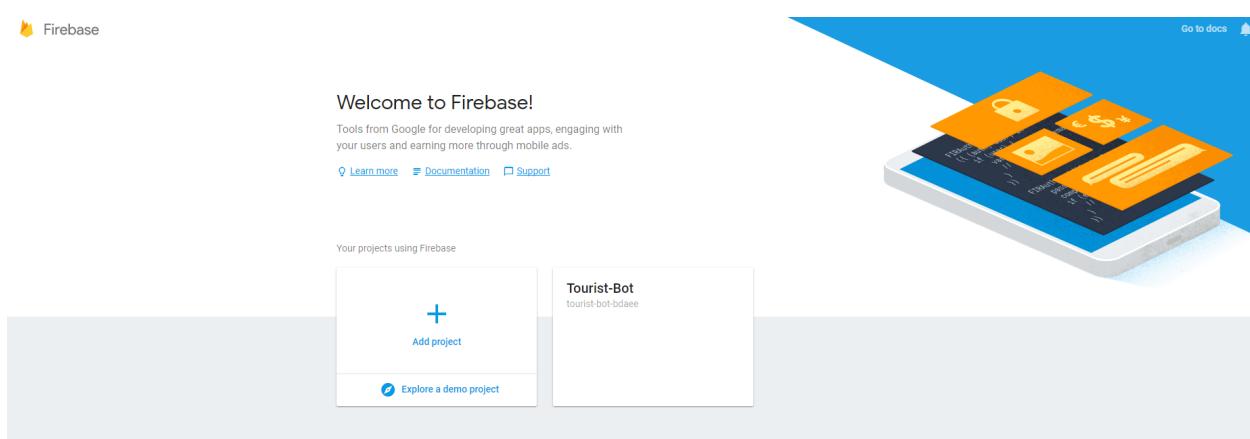


Once you click on **GO TO CONSOLE** you can see all the projects that you have.

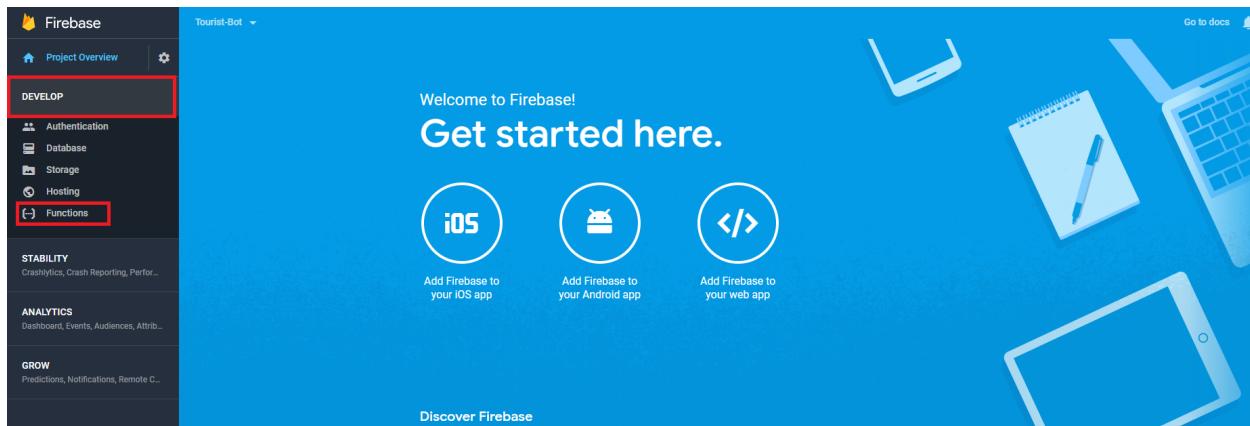
Note : If you create an Agent in Dialogflow, it automatically creates a project in Firebase with the same agent name

Click on your project or create a new project to develop.

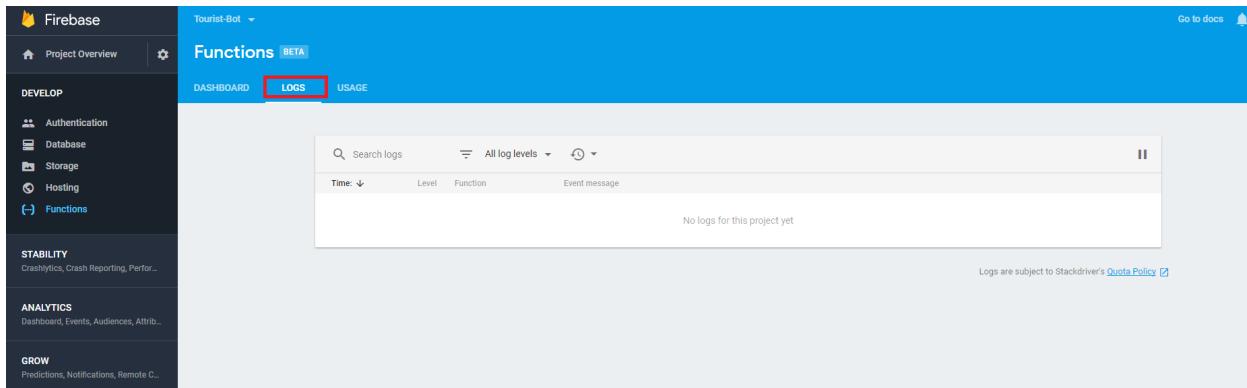
Firebase



Click on **DEVELOP** tile in the left hand side menu and then click on **Functions**.



By default, you will get DASHBOARD. You navigate to **LOGS** where you can view logs and usage of functions in Firebase.



Now click on **DEPLOY** button at the bottom in Dialogflow. Once deployment is done you will get message as "**Successfully deployed your Cloud Function and saved your Webhook**".

This means that, created functions in Inline Editor of Dialogflow are deployed into Firebase.

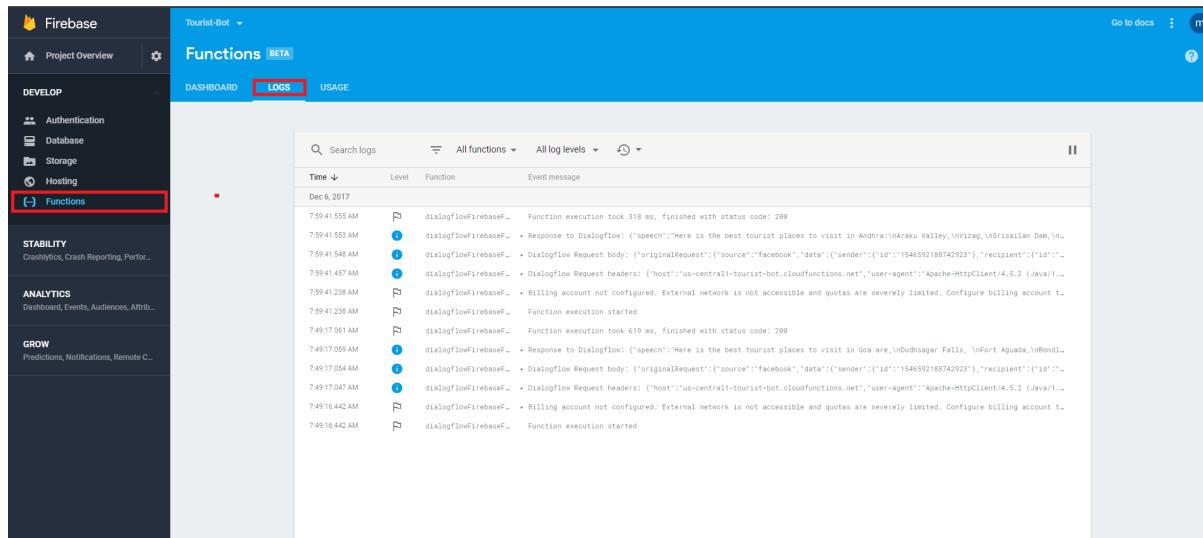
```

index.js
package.json

51         }
52         sendResponse(responseToUser);
53     }
54 }
55 // Default handler for unknown or undefined actions
56 else {
57     let responseToUser = {
58         speech: 'Webhook Request failed', // spoken response
59         text: 'Webhook Request Failed' // displayed response
60     };
61     sendResponse(responseToUser);
62 }
63 }
64 }
65 // If undefined or unknown action use the default handler
66 if (!actionHandlers[action]) {
67     action = 'default';
68 }
69 // Run the proper handler function to handle the request from Dialogflow
70 actionHandlers[action]();
71 // If the response is correctly formatted responses to Dialogflow which are then sent to the user
72 function sendResponse (responseToUser) {
73     // If the response is a string send it as a response to the user
74     if (typeof responseToUser === 'string') {
75         let responseJson = {};
76         responseJson.speech = responseToUser; // spoken response
77         responseJson.displayText = responseToUser; // displayed response
78         responseJson.responseType = 'Simple'; // send response to Dialogflow
79     } else {
80         let responseJson = {};
81         responseJson.speech = responseToUser.speech || responseToUser.displayText;
82         responseJson.displayText = responseToUser.displayText || responseToUser.speech;
83         responseJson.data = responseToUser.data;
84         responseJson.contextOut = responseToUser.outputContexts;
85         responseJson.data = responseToUser.data;
86         responseJson.data = JSON.stringify(responseJson);
87         console.log("Response to Dialogflow: " + JSON.stringify(responseJson));
88         responseJson.responseType = 'Simple'; // send response to Dialogflow
89     }
90 }
91
92
93
94
95
96
97
98

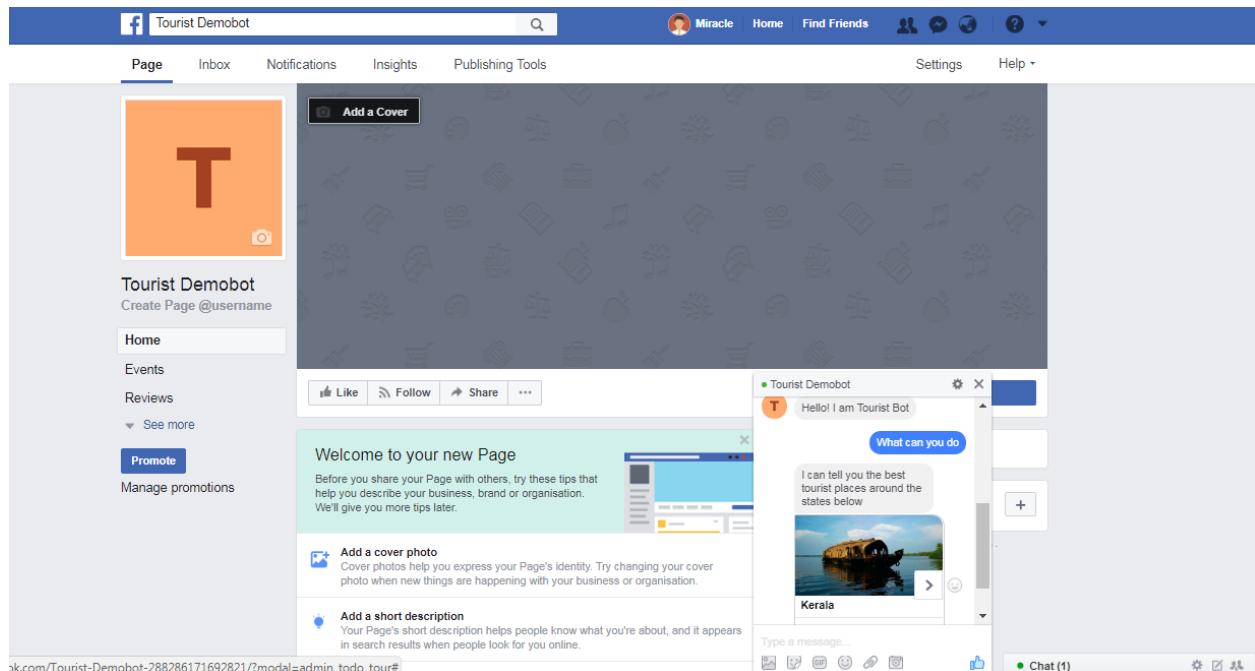
```

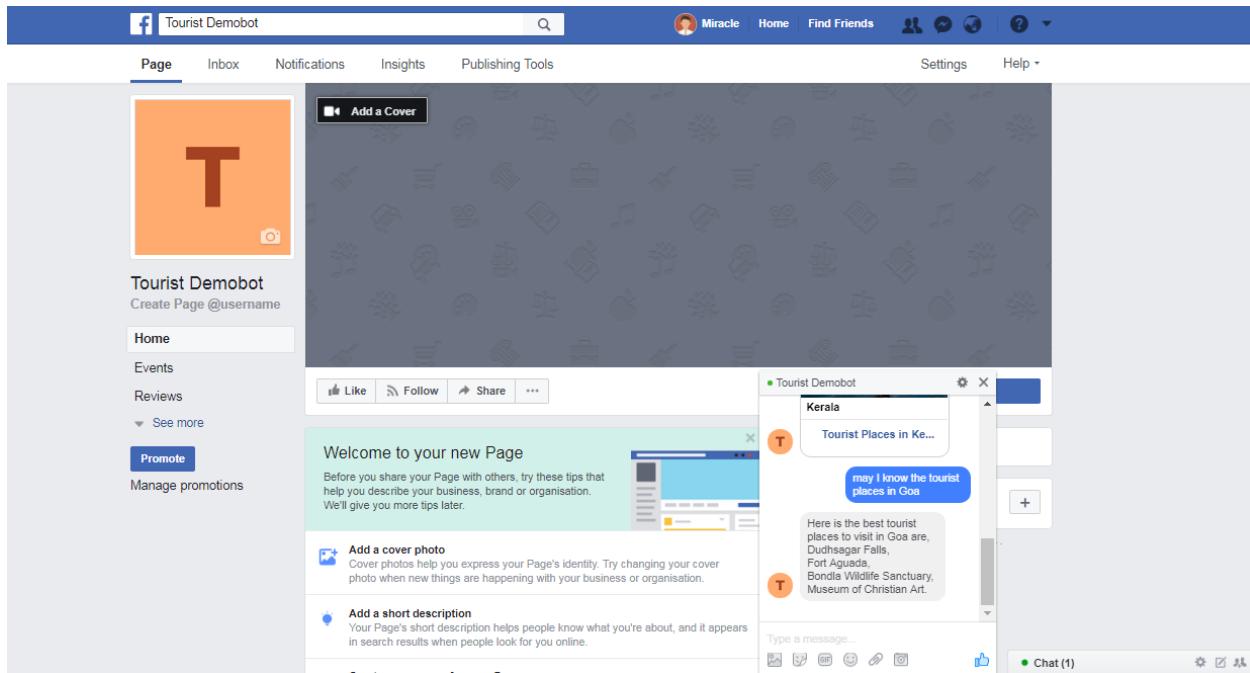
You can check your logs under Functions in Firebase console, once user start interacting with bot <https://console.firebaseio.google.com/>



Testing Tourist Demo Bot in Facebook Messenger

Open your Facebook Messenger and open the chat window of Tourist DemoBot (It is name of the bot in our case and you have to give your app name) and start your conversation.





Hurrah! With this lab you were able to create your first chat bot with 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 😊