

Building a Facebook Messenger Bot with Dialogflow

DS '17 Hands-On Cognitive Lab

Anusha Vennela

Cognitive Research Associate
Miracle Software Systems, Inc.

December 7th, 2017

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
- GitHub repository link : <https://github.com/MiracleLabs/ds17-dialogflow-lab>

Technology Involved

- Dialogflow
- Firebase Functions

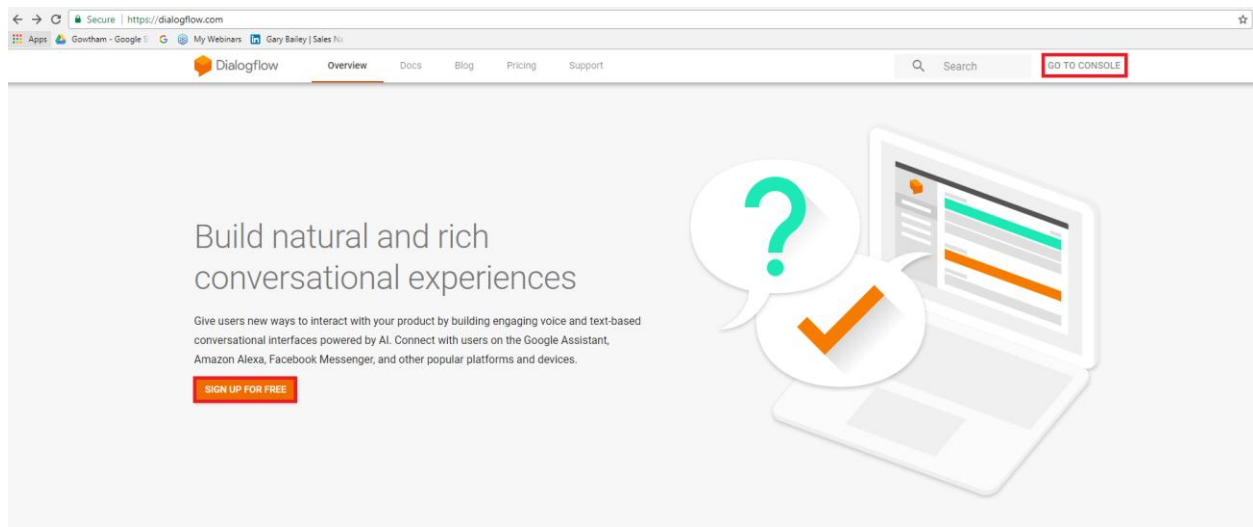
Labs Steps

So, let us get started with the bot!

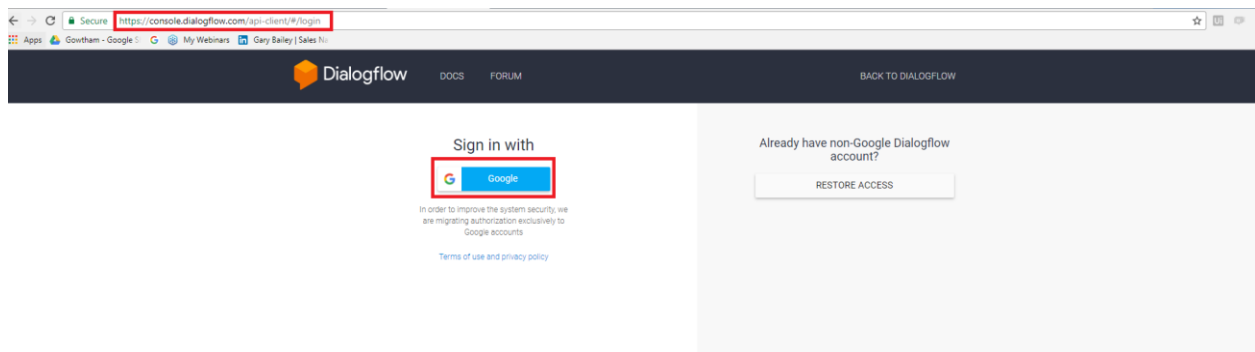
The following steps will outline how you can create a Tourist Bot and integrate it into Facebook Messenger. Users will be able to directly message your bot and perform operations such as "May I know the tourist places to visit in Goa. "

#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**.



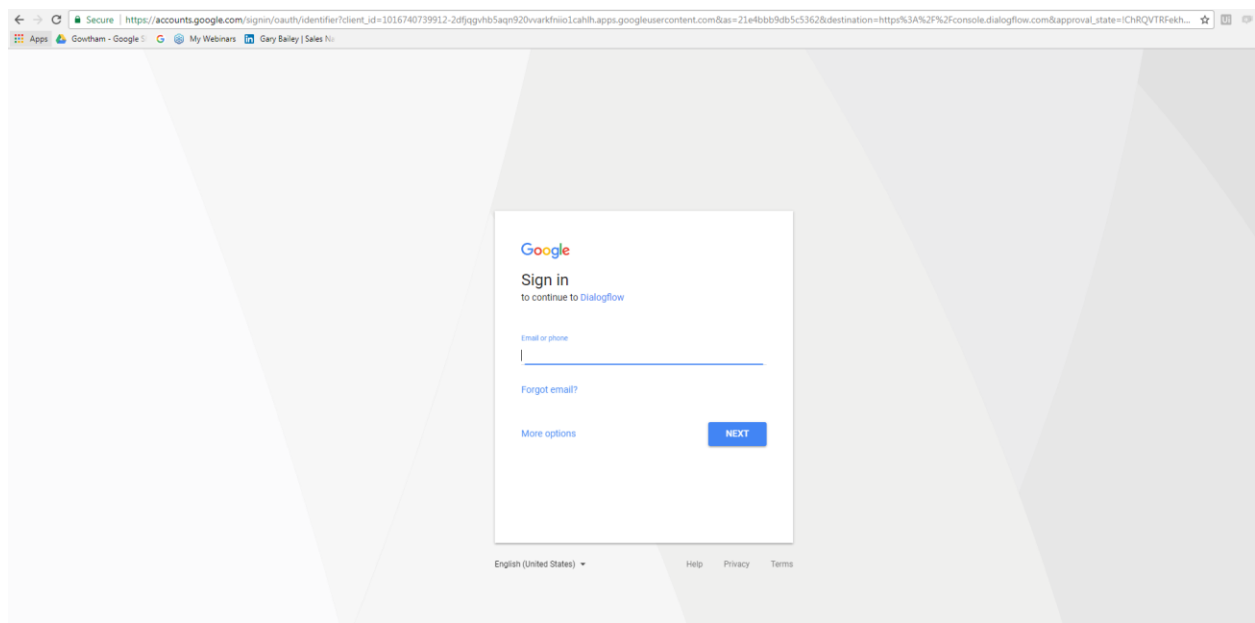
After clicking on any one of the button as above, it will be navigated to the console of Dialogflow for login. Now, click on **Google** button.



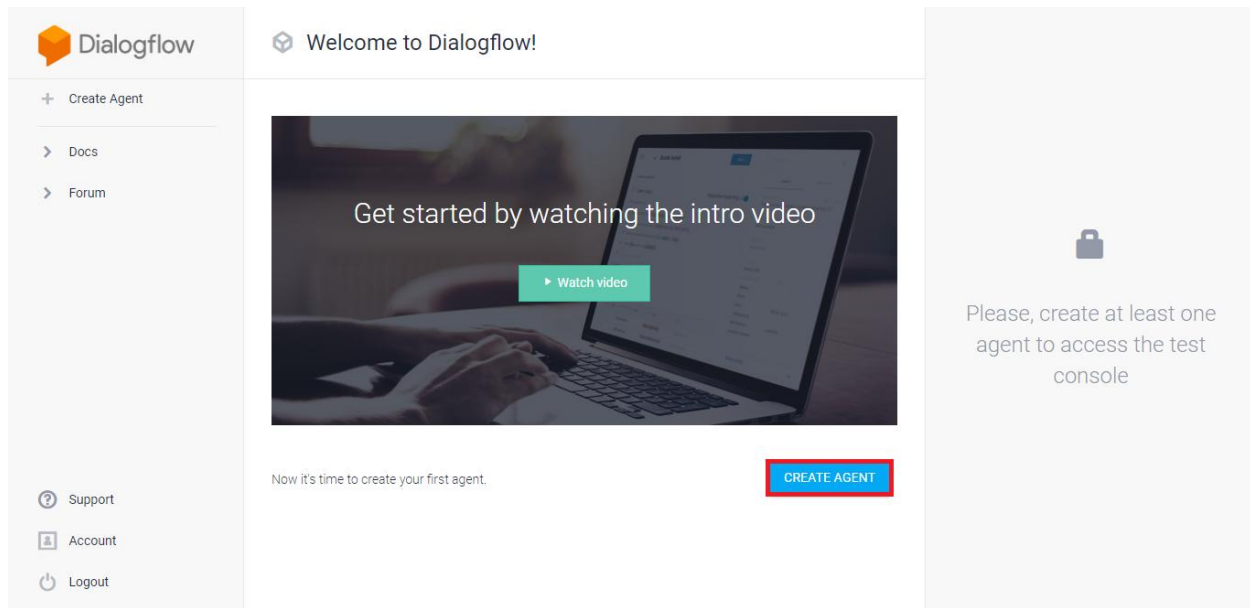
Dialogflow account can be created with Google account credentials. If you have a Google account then enter those credentials and authorize yourself.

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

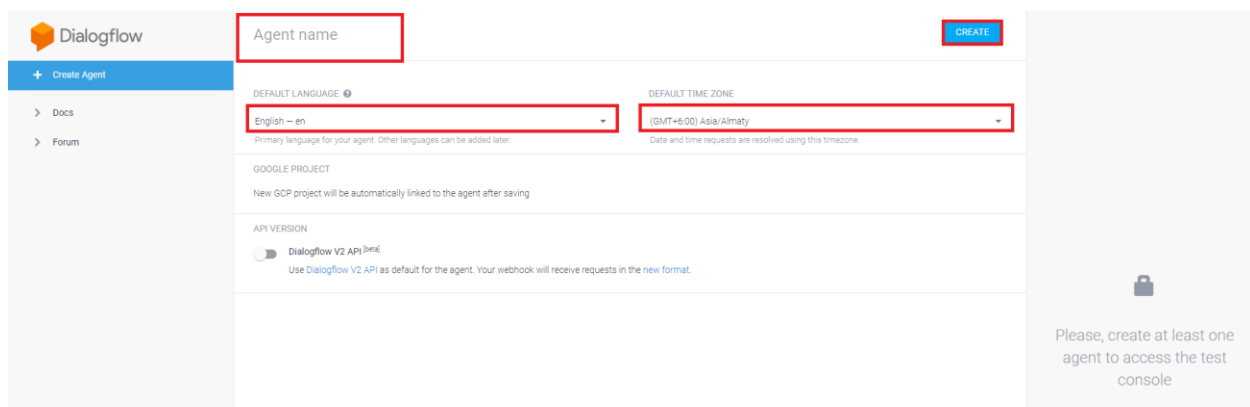
<https://accounts.google.com/SignUpWithoutGmail?hl=en/>



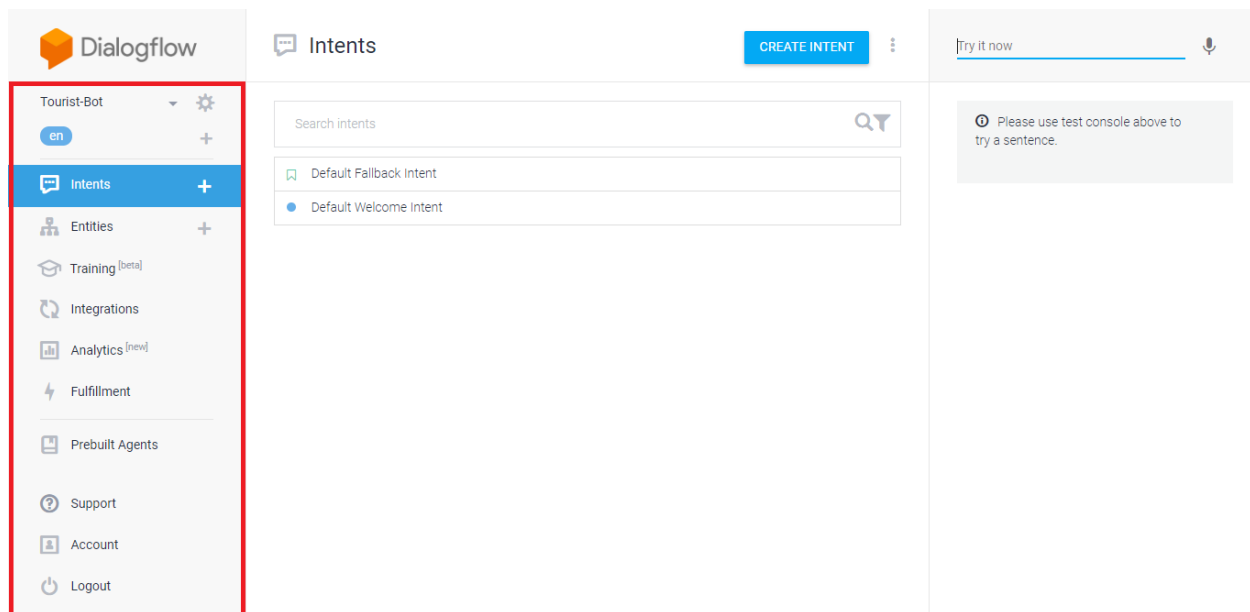
Now, you will get a Dialogflow home page where you have to create an agent/bot. Click on **CREATE AGENT**(if a pop up opens stating don't have correct permissions just Authorize).



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.

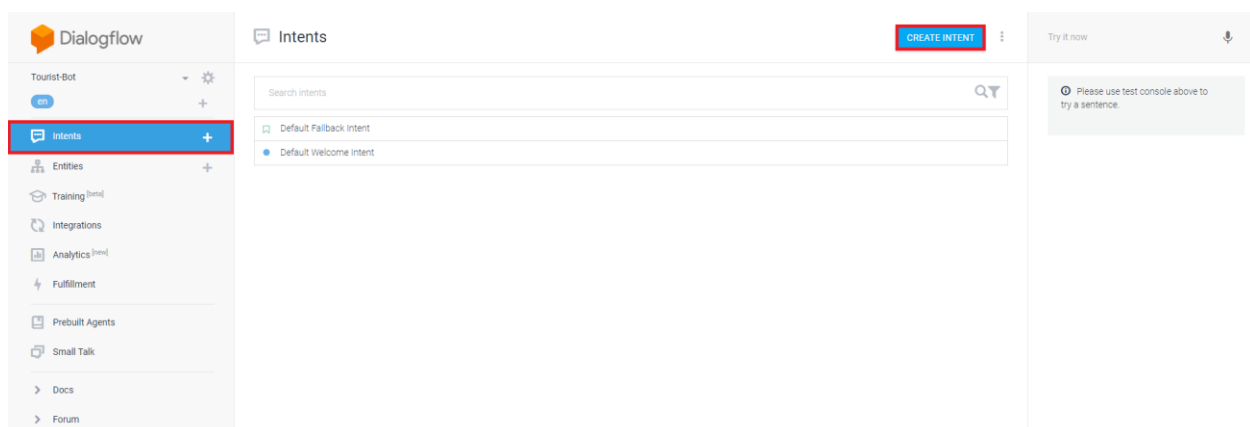


You can read about these in detail through Dialogflow [docs](#).

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.

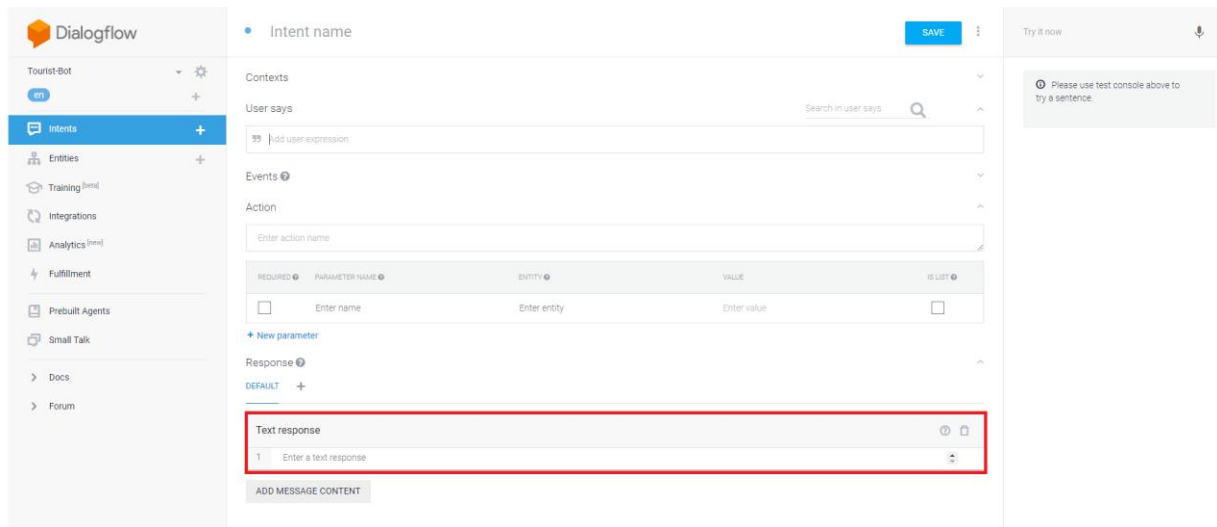
Note : Under User says, as many expressions you provide the Agent will become more matured.

The screenshot shows the Google Dialogflow console interface. On the left is a sidebar with navigation options: Tourist-Bot, Intents, Entities, Training, Integrations, Analytics, Fulfillment, Prebuilt Agents, and Small Talk. The main area is titled 'Intent name' and has a 'SAVE' button. Below this, there are sections for 'Contexts', 'User says' (with a search bar and a red box around the 'Add user expression' button), 'Events', 'Action' (with a table for parameters), and 'Response'. The 'Response' section is expanded, showing a 'Text response' with a single example: 'Enter a text response'.

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
<input type="checkbox"/>	Enter name	Enter entity	Enter value	<input type="checkbox"/>

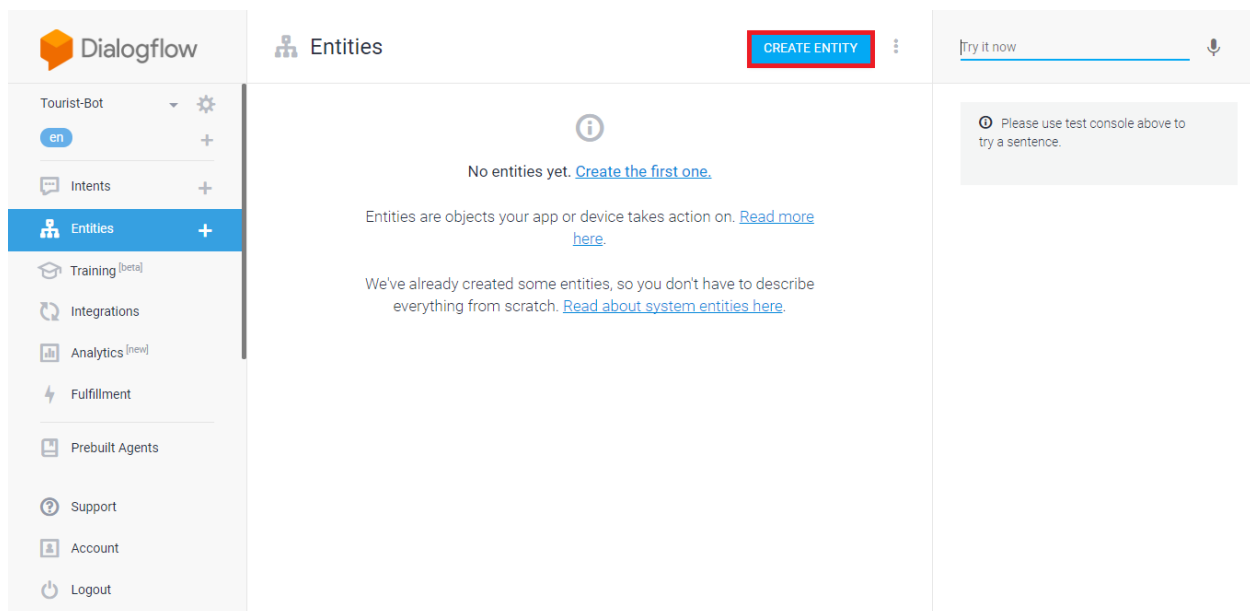
Below the table, there is a '+ New parameter' link. The 'Response' section is expanded, showing a 'Text response' with a single example: 'Enter a text response'.

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.

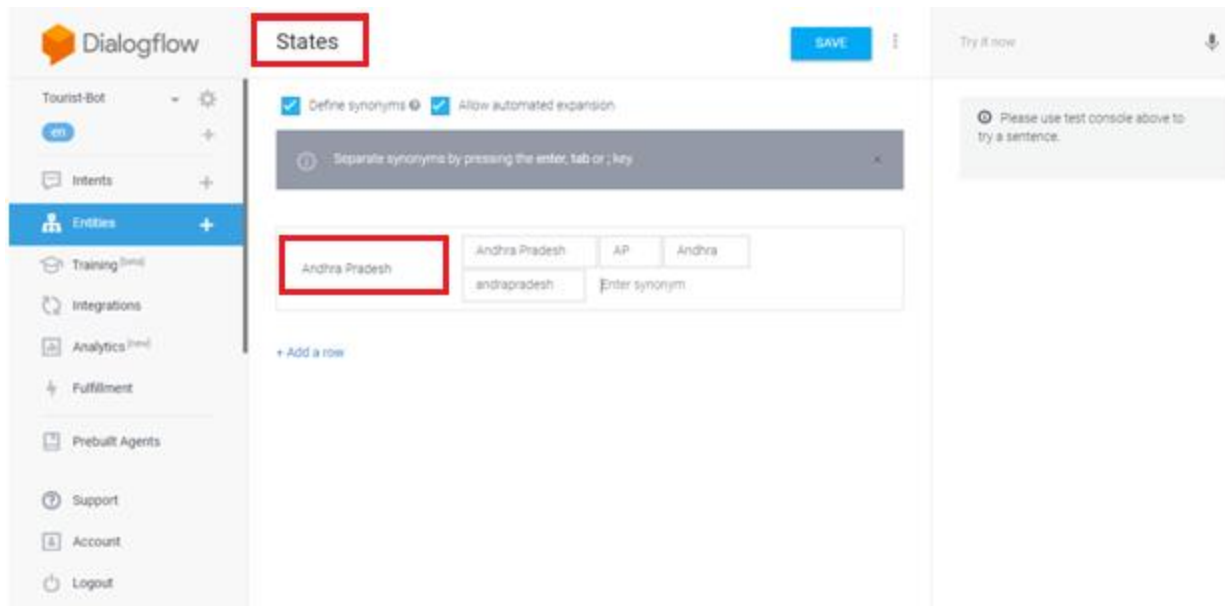


Creation of Entities

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.

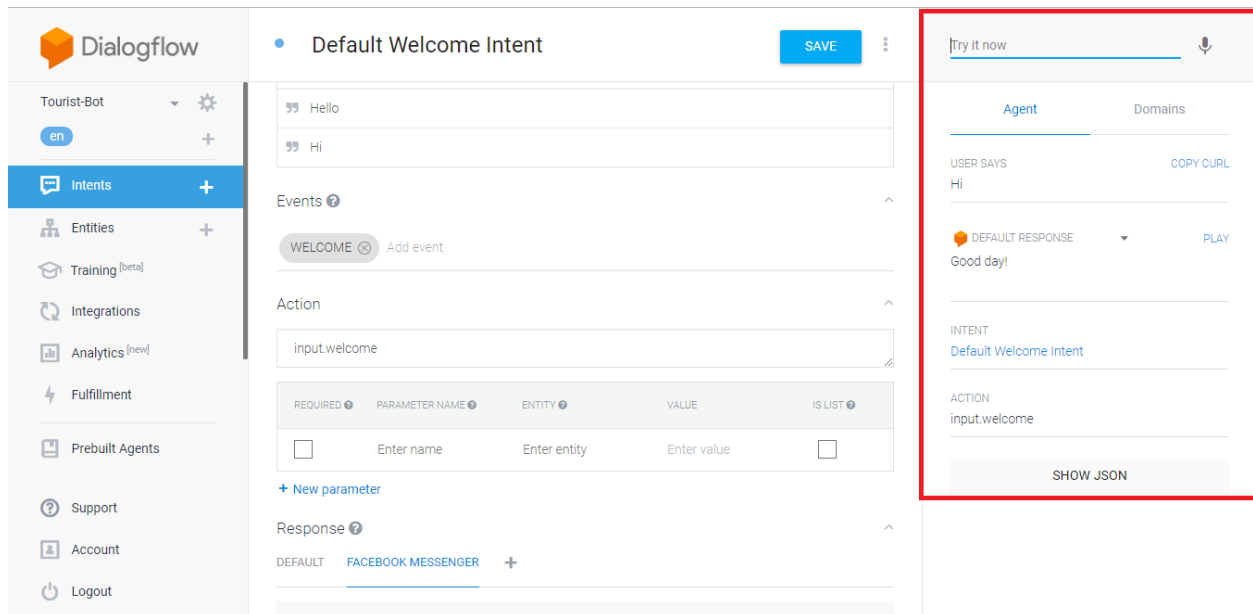


You need to give the entity name and also you have to provide the reference value and synonyms for it while creating the Intent as following,



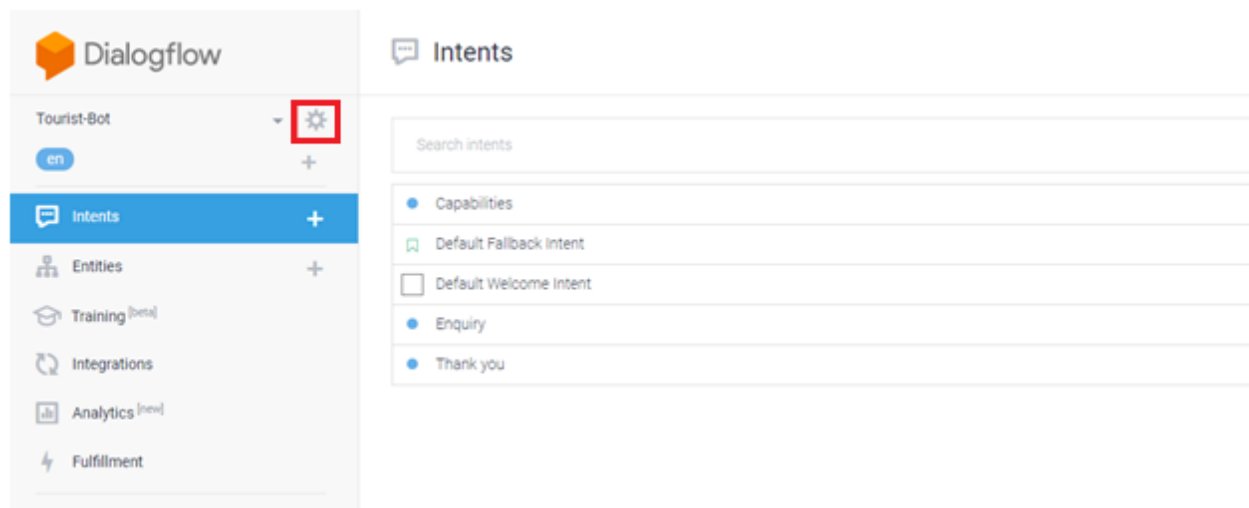
Then click on **SAVE** button on the top right to save the created entity.

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

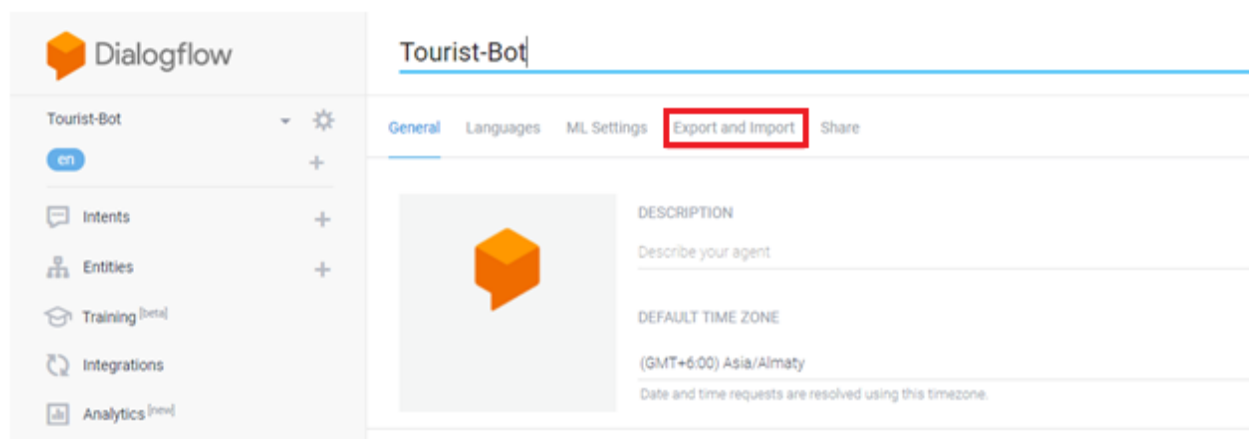


You can find the sample Agent in the GitHub repository <https://github.com/MiracleLabs/ds17-dialogflow-lab> which is created with the name **Tourist-Bot.zip** and download that zip file in your system.

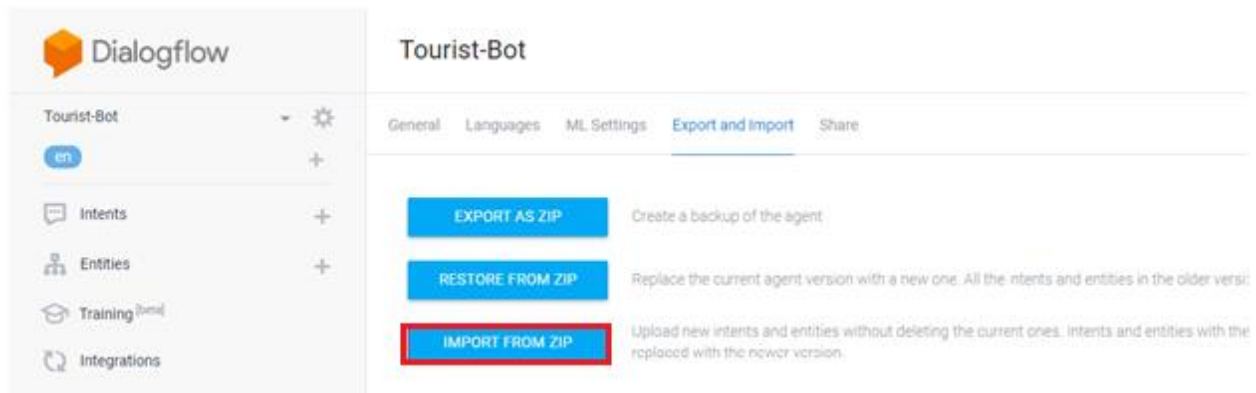
Now go to Dialogflow console and click on gear icon as shown below to import Intents and Entities.



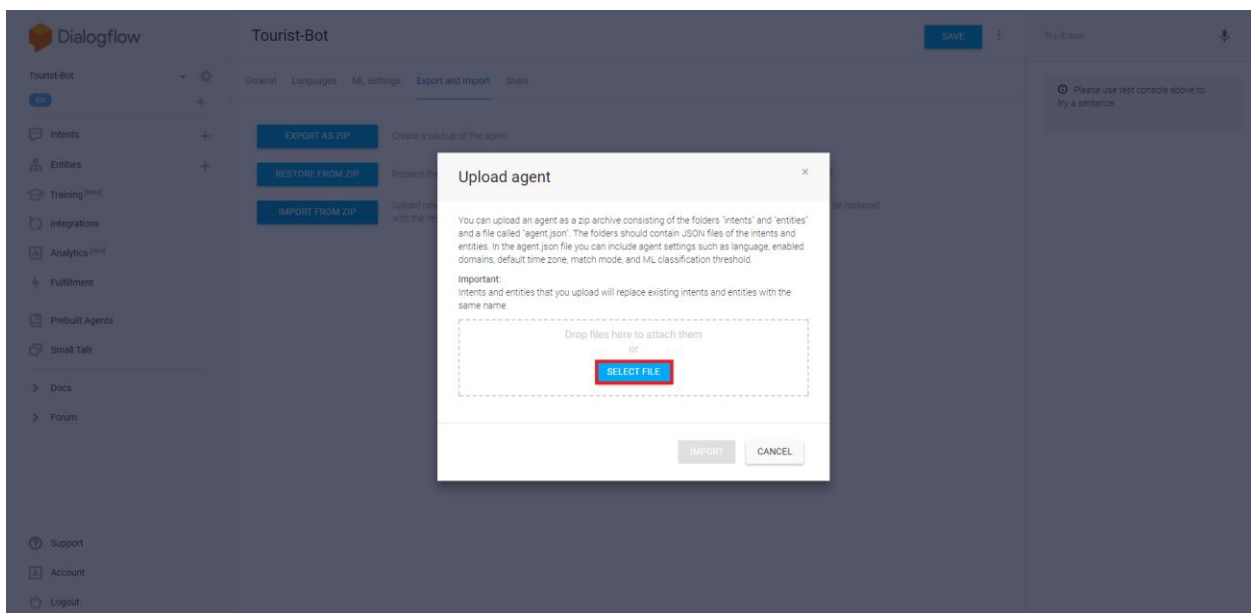
By default, you will be landed on General session page. Select **Export and Import** tab.



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



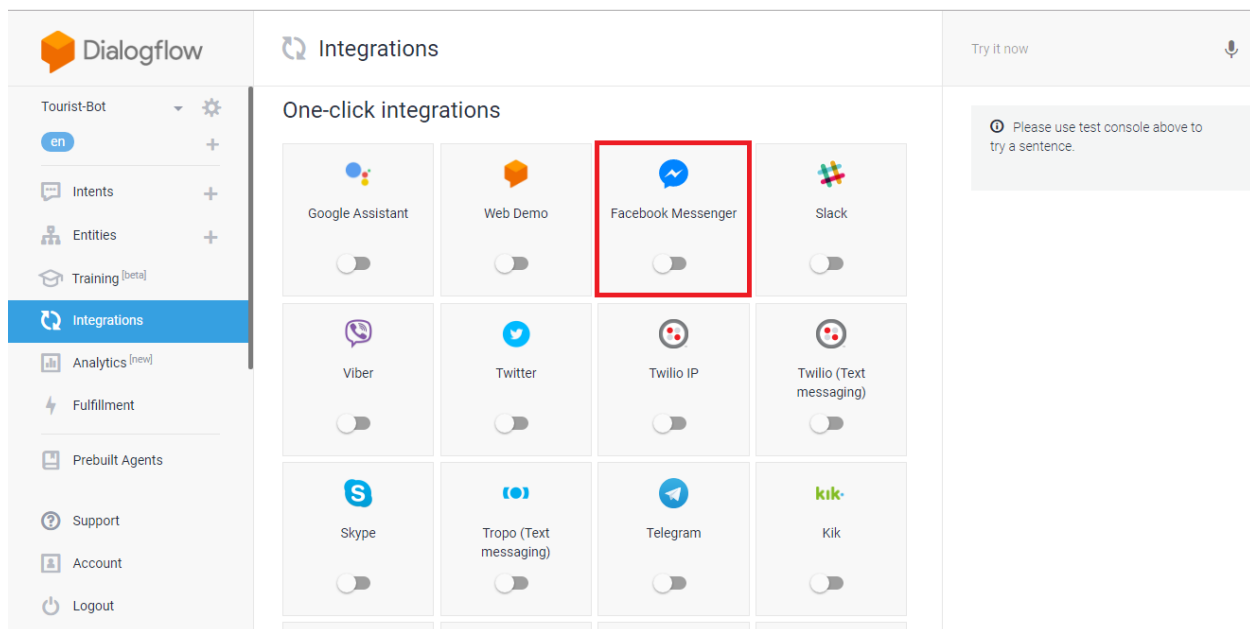
Now you will get a pop-up for asking you to **SELECT FILE**, you have to upload the zip file which you downloaded previously from GitHub repository. Once upload is done click on **IMPORT** button



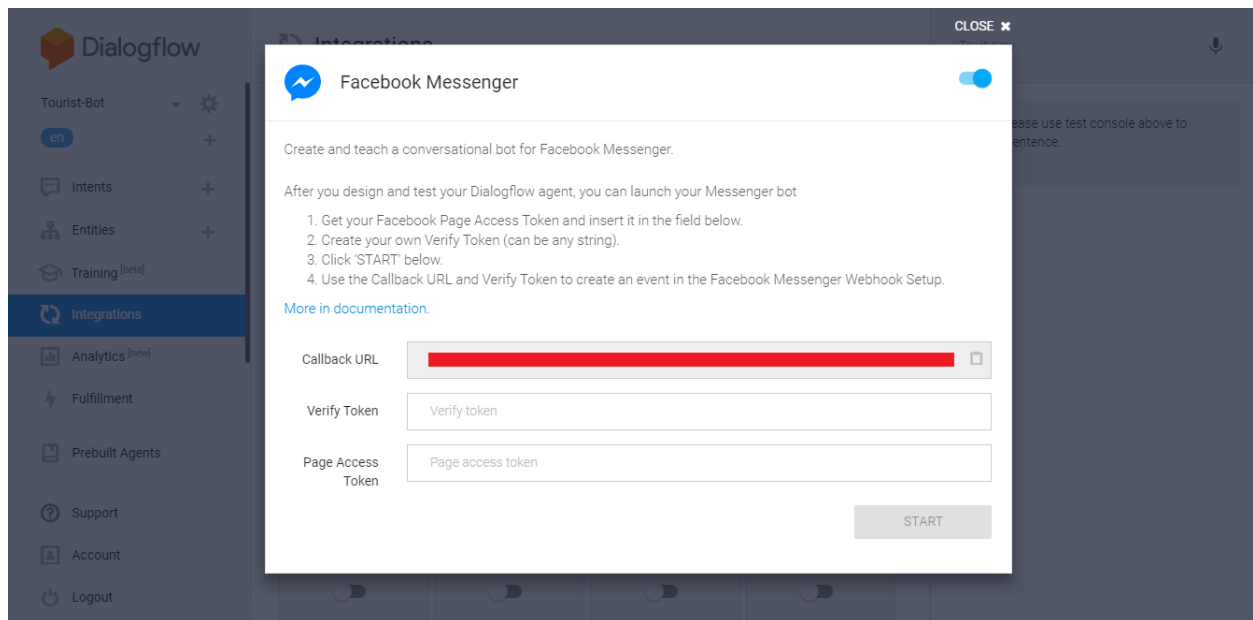
Once importing zip file is done you will find the agent of Tourist Bot which you imported.



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,

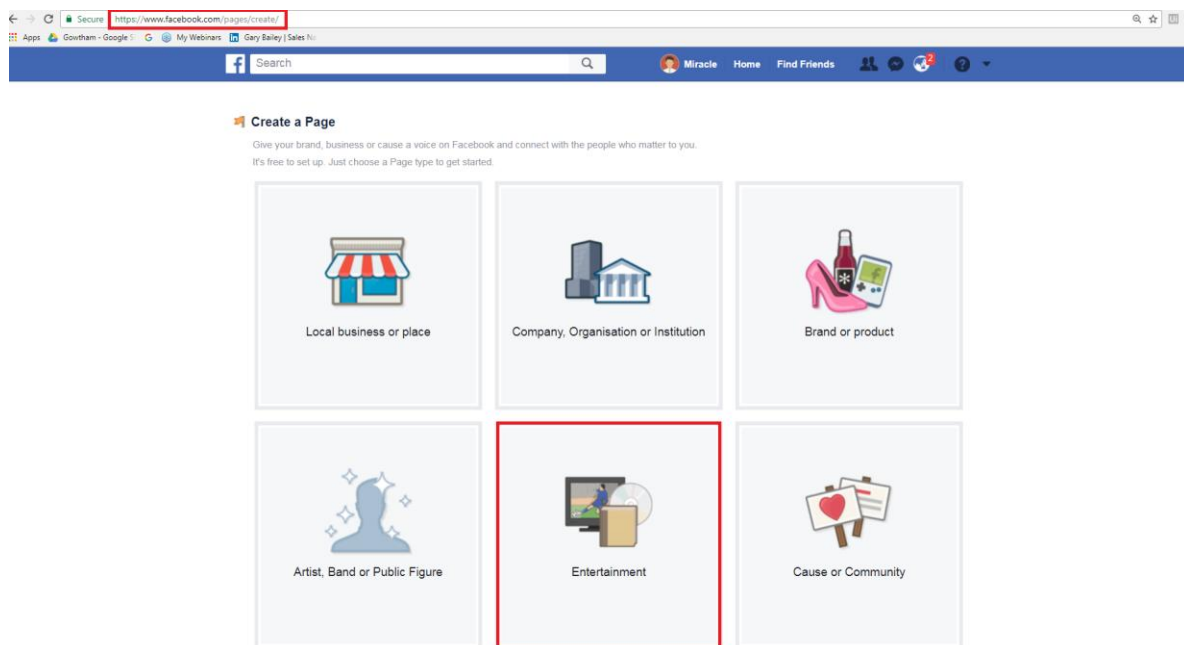
<https://www.facebook.com/>



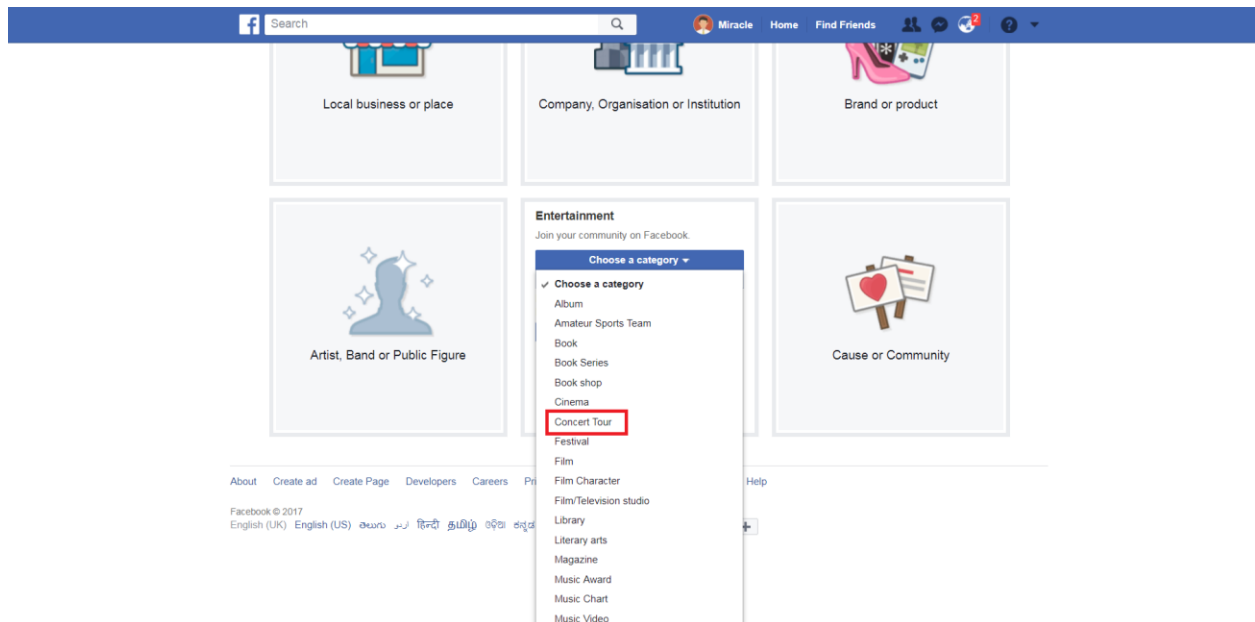
Once you login into Facebook, you need to create a new page by opening the below link in a new tab

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

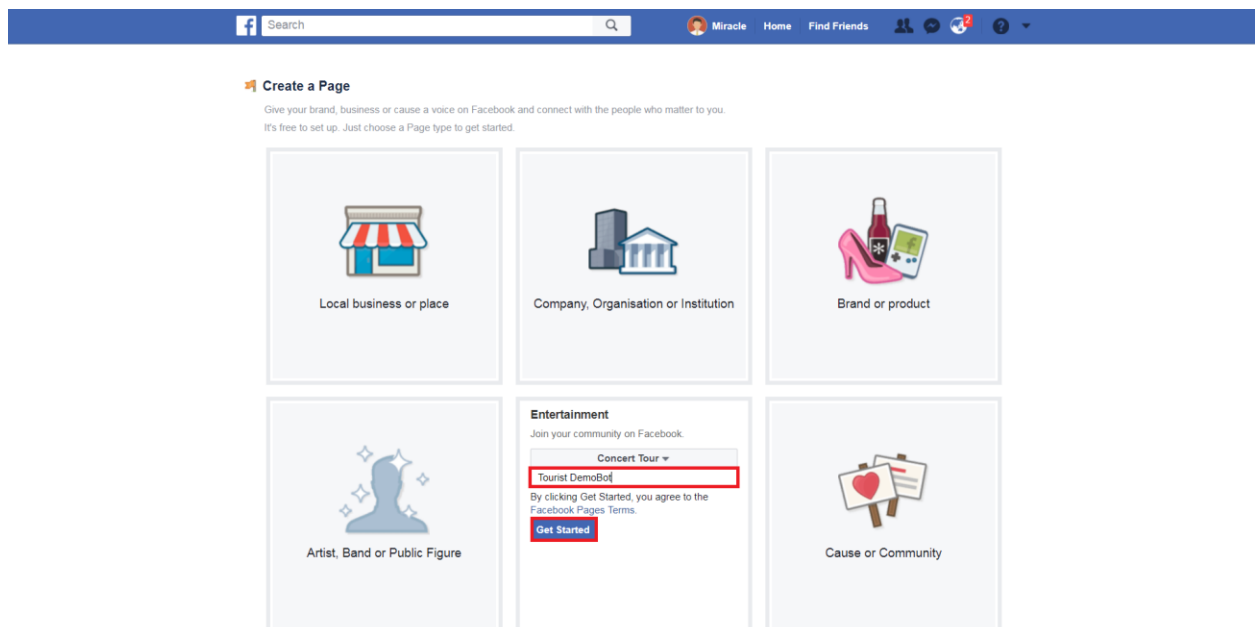
You need to select page type to get started. As Tourist Bot comes under entertainment category, select the **Entertainment** tile.



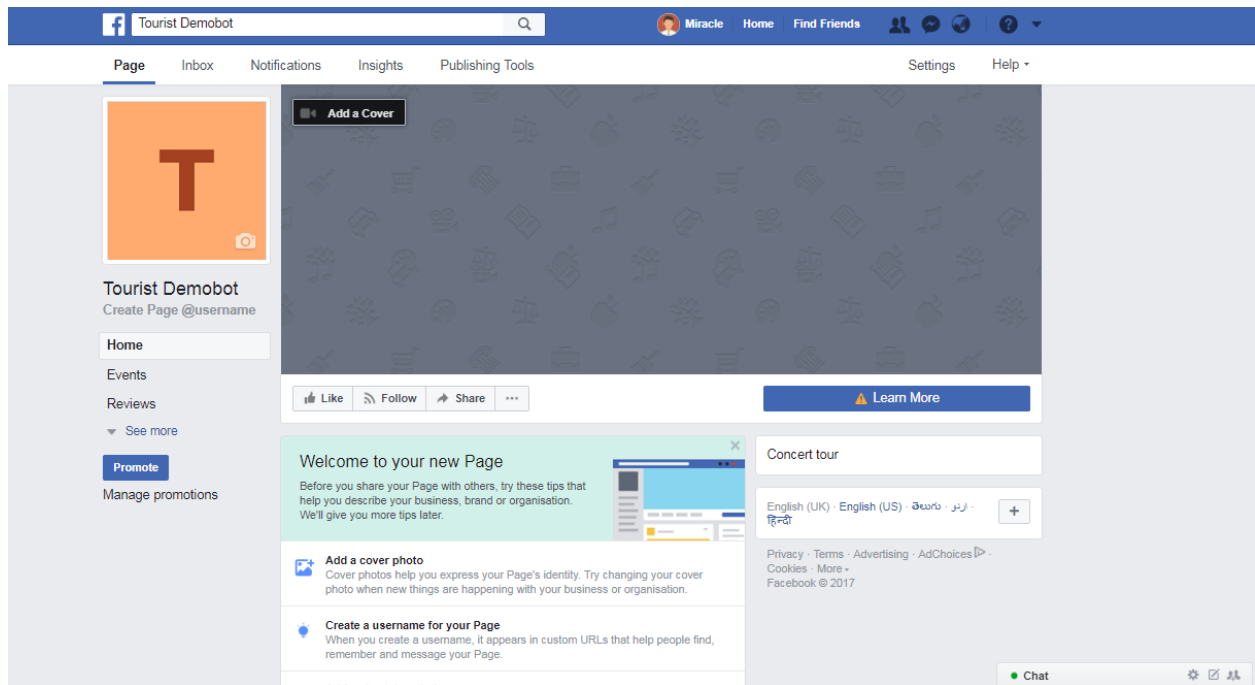
Now, from the dropdown list select **Concert Tour** under Choose a Category.



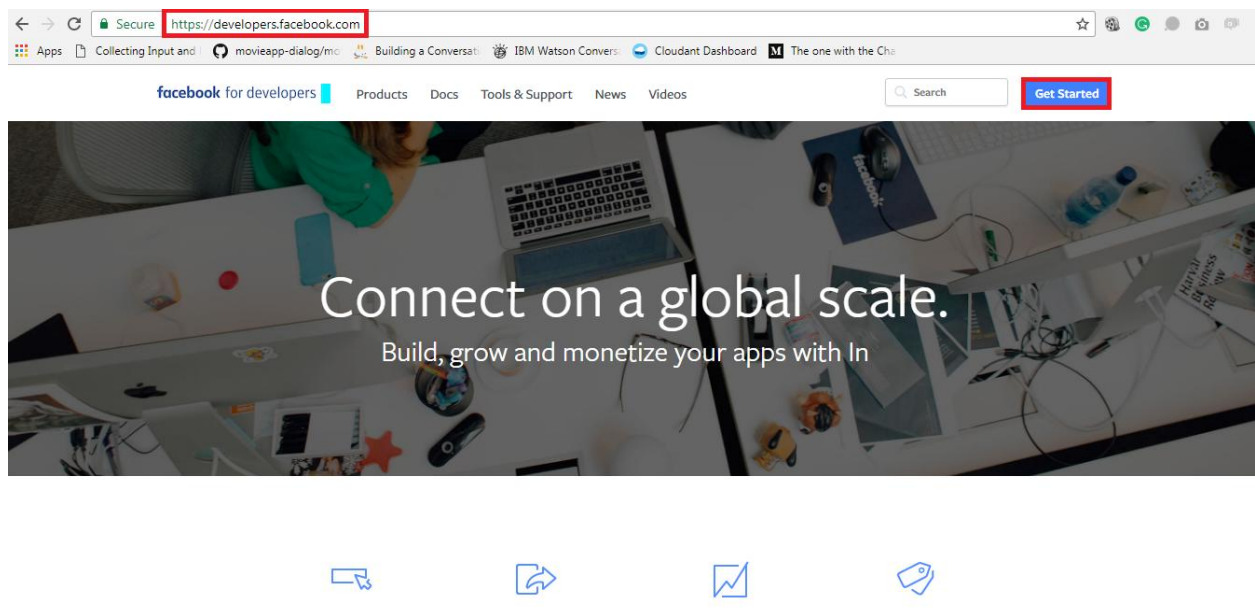
You need to provide a **Name** for creating a page. For Example: <Tourist DemoBot> and click on **Get Started**.



Your page will be created and looks like below,

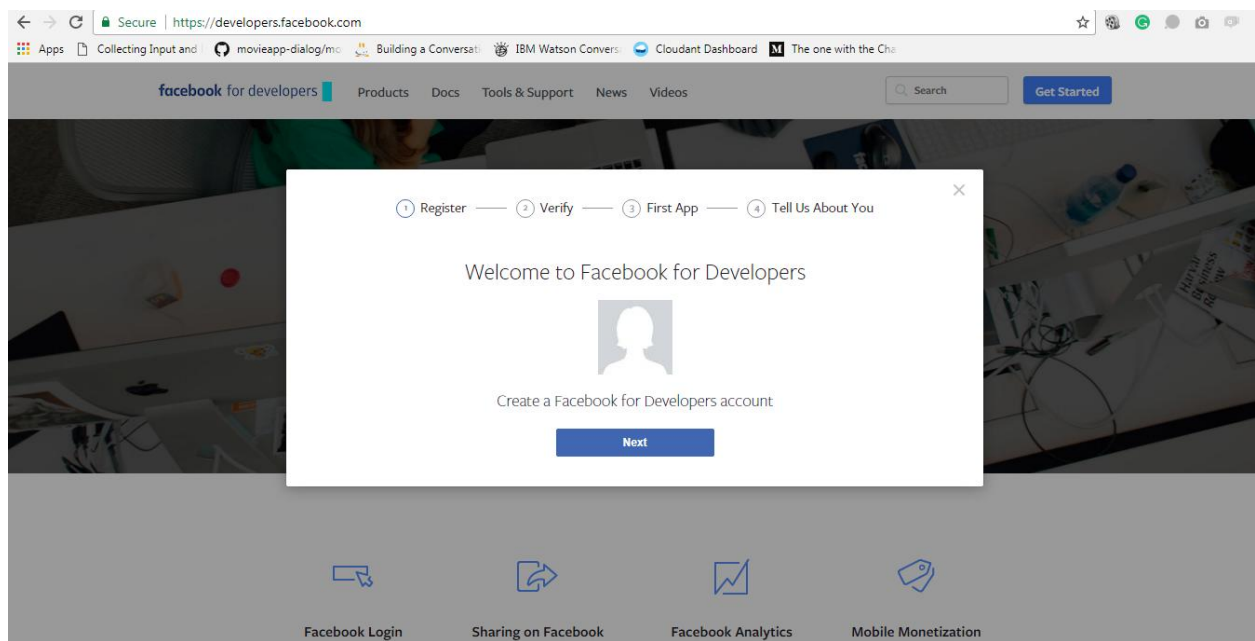


After your page is created, go to the <https://developers.facebook.com/> in a new tab and Click on **Get Started**

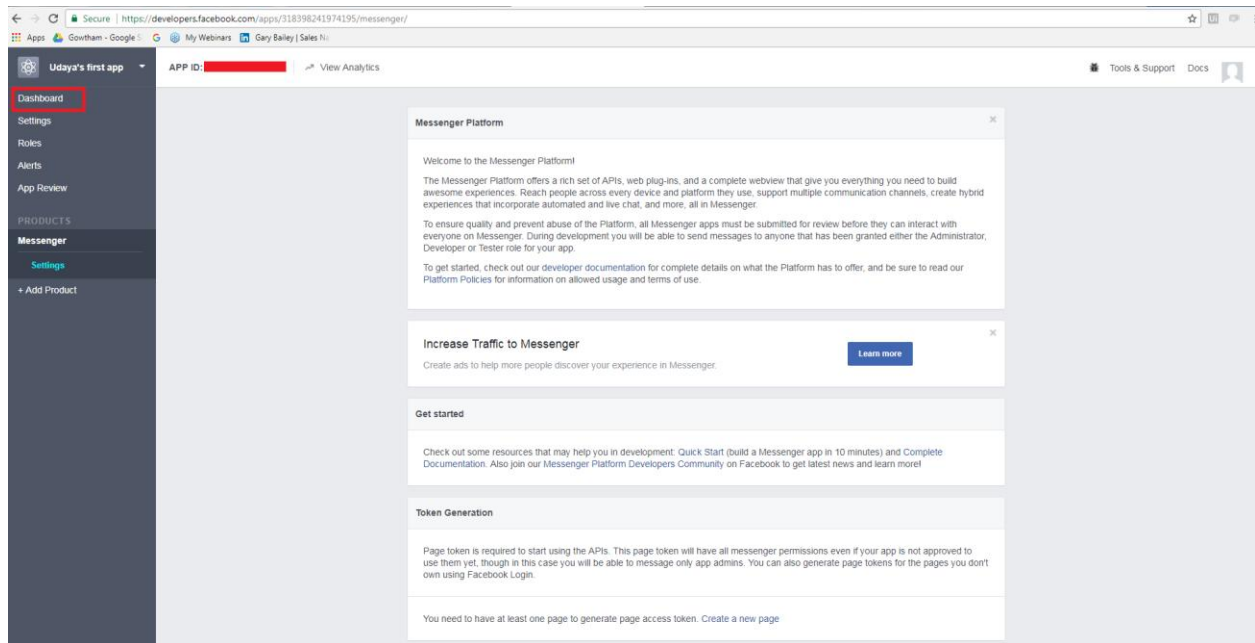


Complete all the 4 steps - **Register, Verify, First App** and **Tell Us About You**. For these steps, authenticate yourself by Email or Phone and provide <Your App Name> Example: “Miracle's Tourist Bot” as the app name.

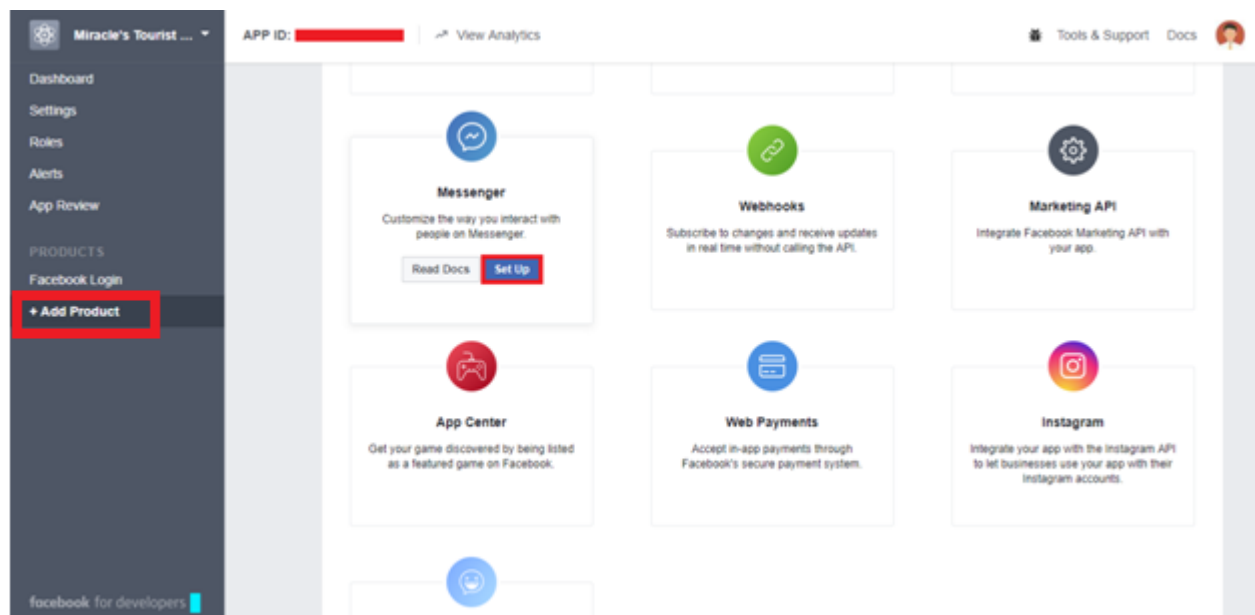
Note : By default, it gives one app name and if you want to go with the app name which you are getting by default then follow the next steps. Orelse rename the app name.



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



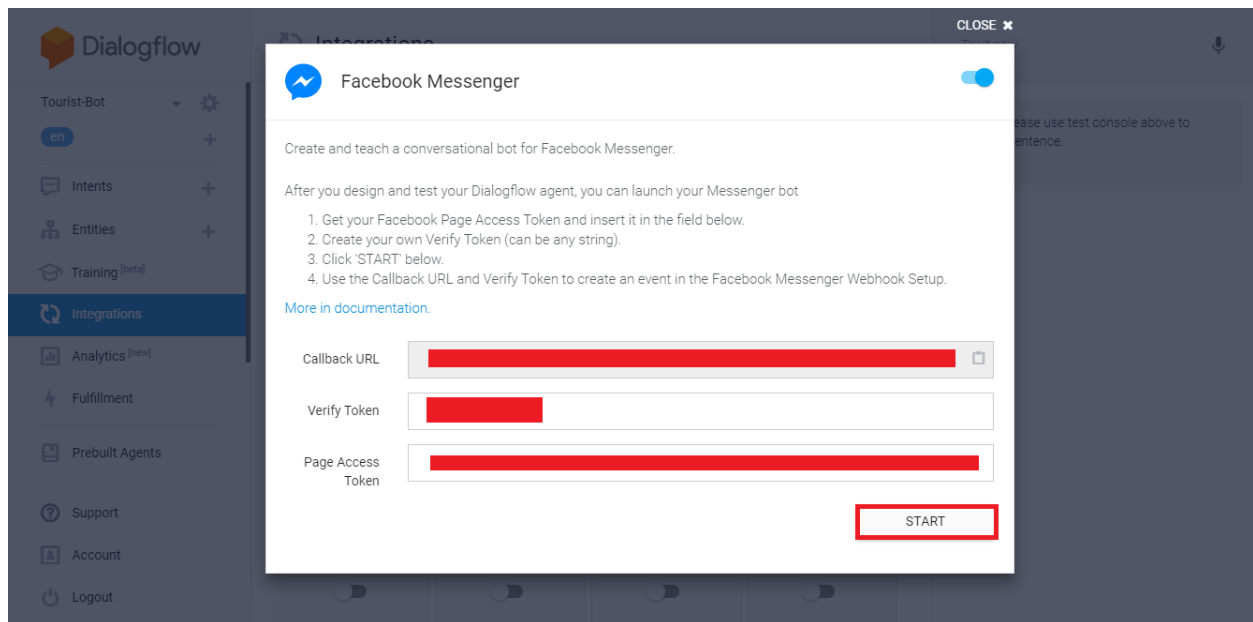
Click on **+ Add Product** Choose Messenger and then click on **Set Up**.



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.

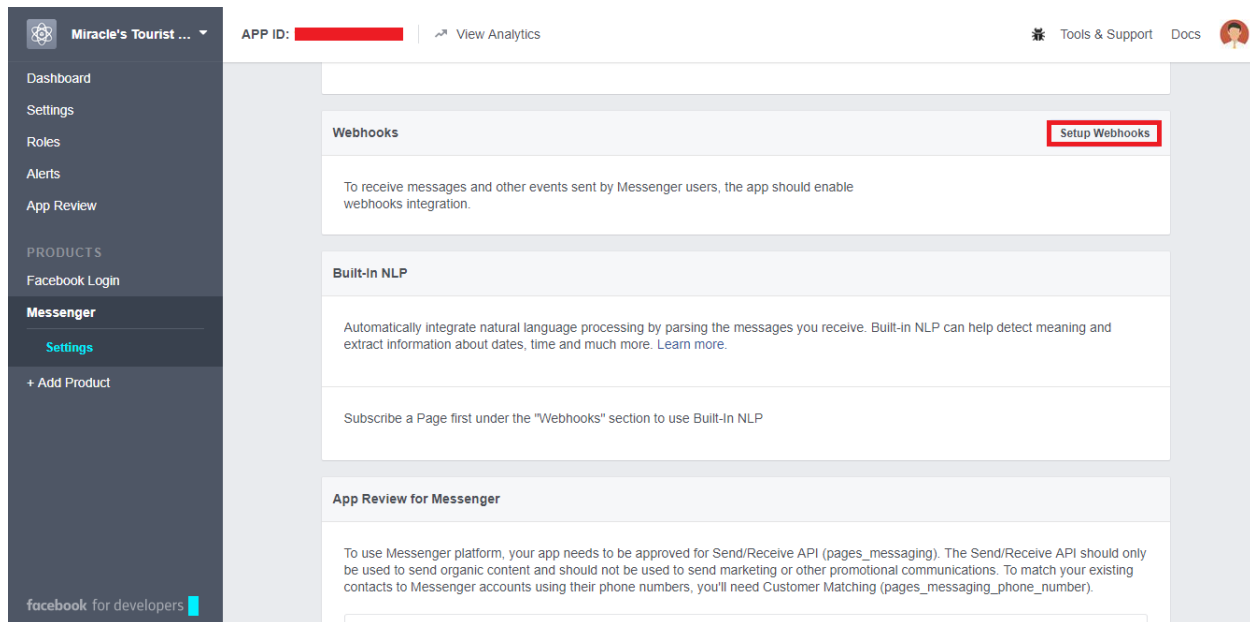
The screenshot shows the Facebook Developer console interface. On the left is a sidebar with navigation links: Dashboard, Settings, Roles, Alerts, App Review, PRODUCTS, Facebook Login, Messenger (highlighted), Settings (sub-link), and + Add Product. The main content area is titled 'Miracle's Tourist ...' and shows the 'Token Generation' section. It includes a 'Get started' section with links to documentation and a Facebook community group. Below this, the 'Token Generation' section explains that a page token is required and provides instructions on how to generate one. A dropdown menu for 'Page' is set to 'Tourist Demobot', and a red bar represents the 'Page Access Token'. A 'Webhooks' section is also visible at the bottom, with a 'Setup Webhooks' button.

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.

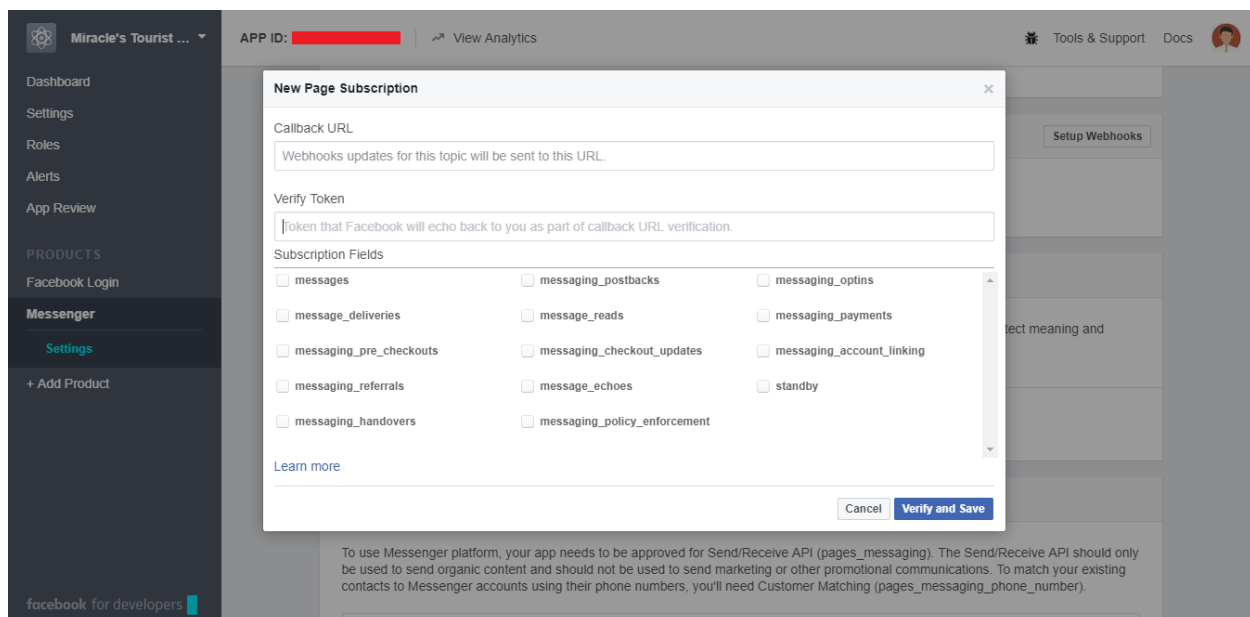


On successful start you will see a message in the bottom right of screen with the message "BOT started successfully".

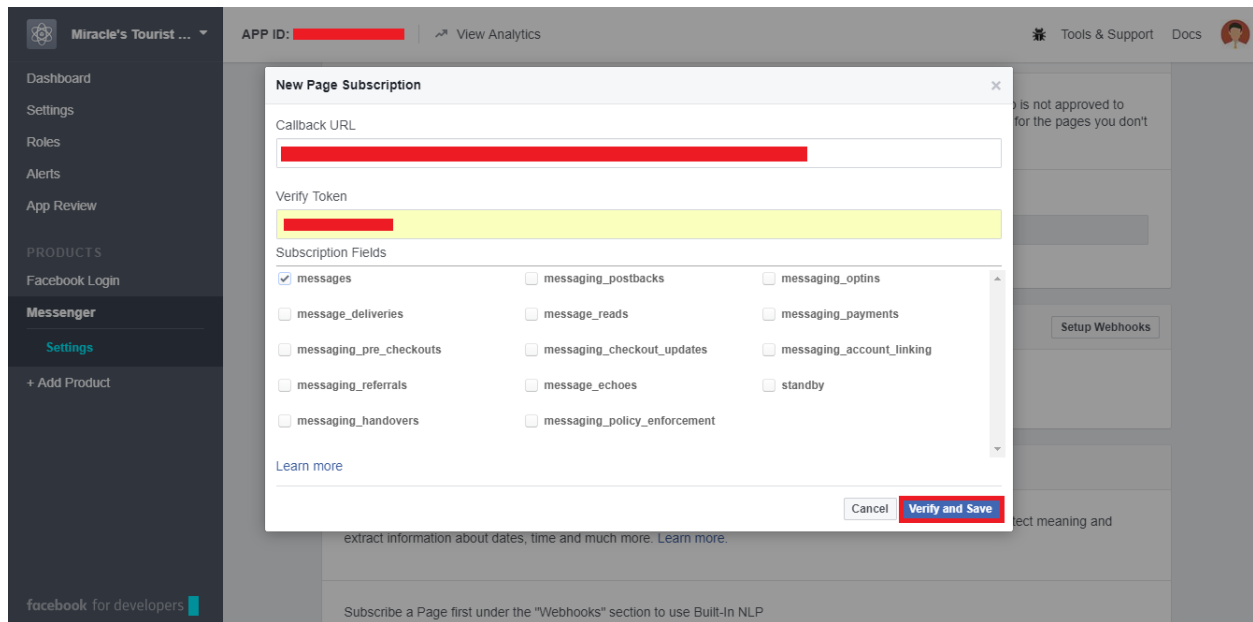
Now, go back to Facebook Developer page and this time you need to add Webhook. So click on **Setup Webhooks** under Webhooks in Messenger Settings.



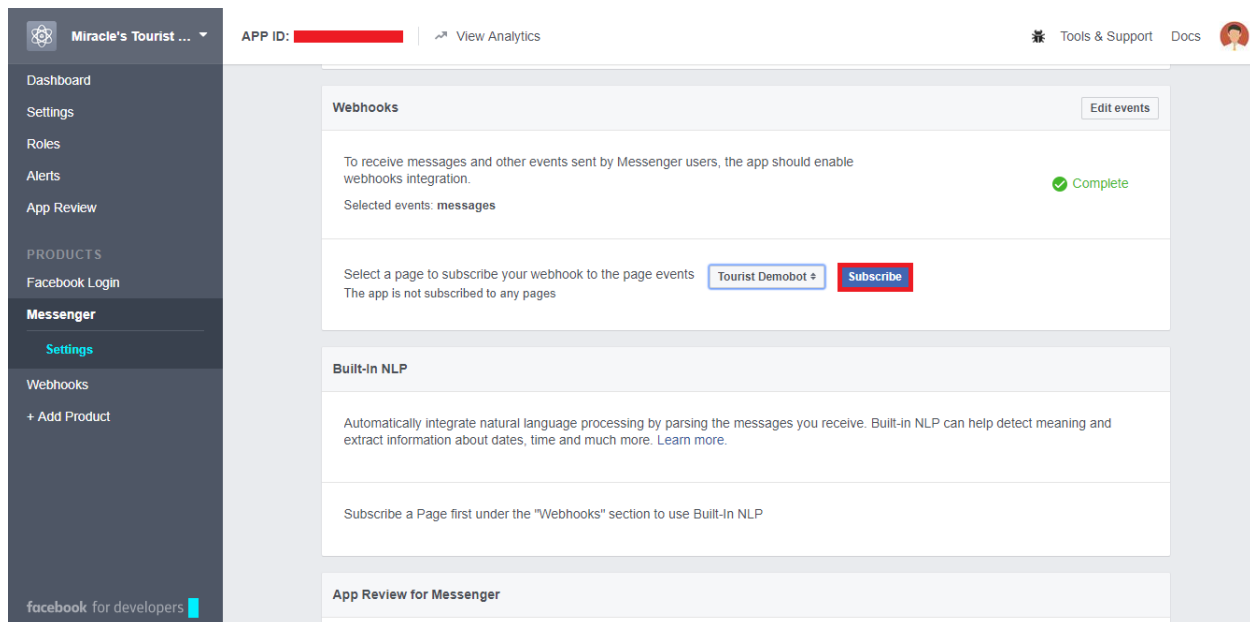
Now, you will get a pop-up like below, which is Webhooks landing page for new page subscription.



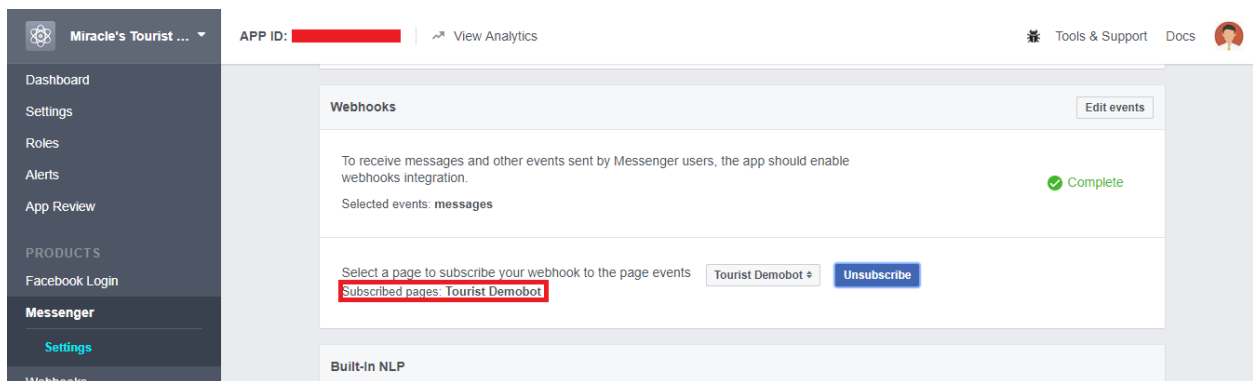
Copy the Callback URL and Verify Token from Dialogflow's Facebook Messenger tile and paste them in the respective fields and choose the options from Suggestion Fields and click on **Verify and Save**.



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



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



#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.

Import the required modules for Fulfillment

```
// Cloud Functions for Firebase library
const functions = require('firebase-functions');
exports.dialogflowFirebaseFulfillment = functions.https.onRequest((request,
response) => {
  console.log('Dialogflow Request headers: ' + JSON.stringify(request.headers));
  console.log('Dialogflow Request body: ' + JSON.stringify(request.body));
});
```

Defining variables

```
function processV1Request(request, response) {
  let action = request.body.result.action;
  // https://dialogflow.com/docs/actions-and-parameters
  let parameters = request.body.result.parameters;
  // https://dialogflow.com/docs/contexts
```



```
let inputContexts = request.body.result.contexts;
let requestSource = (request.body.originalRequest) ?
request.body.originalRequest.source : undefined;
}
```

Create handlers for Dialogflow actions as well as a 'action' handler

```
const actionHandlers = {
  'Places': () => {
    if (request.body.result.parameters.States == "Andhra Pradesh") {
      // Send simple response to user
      let responseToUser = {
        // spoken response
        speech: "The most visited tourist places in Andhra are:\n" + "Araku Valley,\n"
+ "Vizag,\n" + "Srisailan Dam,\n" + "Papikondalu,\n" + "Talakona Waterfalls.\n",
        // displayed response
        text: 'This is from Dialogflow\'s Cloud Functions for Firebase editor! '
      };
      sendResponse(responseToUser);
    }
  }
}
```

Function to send correctly formatted responses to Dialogflow

```
function sendResponse(responseToUser) {
  // if the response is a string send it as a response to the user
  if (typeof responseToUser === 'string') {
    let responseJson = {};
    responseJson.speech = responseToUser; // spoken response
    responseJson.displayText = responseToUser; // displayed response
    response.json(responseJson); // Send response to Dialogflow
  } else {
    let responseJson = {};
    responseJson.speech = responseToUser.speech || responseToUser.displayText;
```

```
responseJson.displayText = responseToUser.displayText ||  
responseToUser.speech;  
responseJson.data = responseToUser.data;  
responseJson.contextOut = responseToUser.outputContexts;  
console.log('Response to Dialogflow: ' + JSON.stringify(responseJson));  
response.json(responseJson); // Send response to Dialogflow  
}  
}
```

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

Below screenshot indicates **package.json**, which contains metadata of our application.

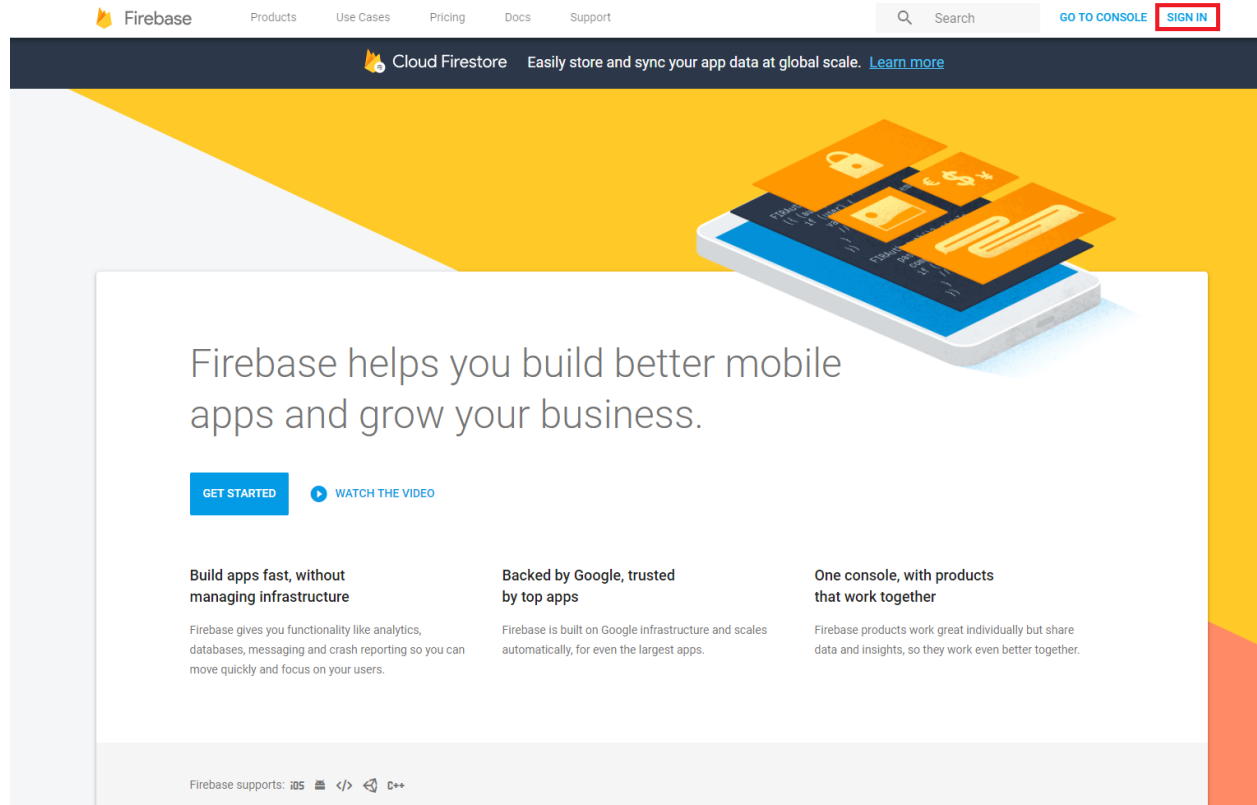
The screenshot displays the Dialogflow Fulfillment Inline Editor interface. On the left is a sidebar with navigation options: Tourist-Bot, Intents, Entities, Training [beta], Integrations, Analytics [new], Fulfillment (selected), Prebuilt Agents, Support, Account, and Logout. The main area is titled 'Fulfillment' and 'Inline Editor (Powered by Cloud Functions for Firebase)'. It shows a file explorer with 'index.js' and 'package.json' (highlighted with a red box). The 'package.json' file is open, displaying the following JSON metadata:

```
{  
  "name": "dialogflowfirebasefulfillment",  
  "description": "This is the fulfillment for a Dialogflow agents using Cloud Functions for F",  
  "version": "0.0.1",  
  "private": true,  
  "license": "Apache Version 2.0",  
  "author": "Google Inc.",  
  "engines": {  
    "node": ">=6.0"  
  },  
  "scripts": {  
    "start": "firebase serve --only functions:dialogflowfirebasefulfillment",  
    "deploy": "firebase deploy --only functions:dialogflowfirebasefulfillment"  
  },  
  "dependencies": {  
    "firebase-admin": "^4.2.1",  
    "firebase-functions": "^0.5.7",  
    "apiai": "^4.0.3",  
    "express": ">=4",  
    "body-parser": ">=1"  
  }  
}
```

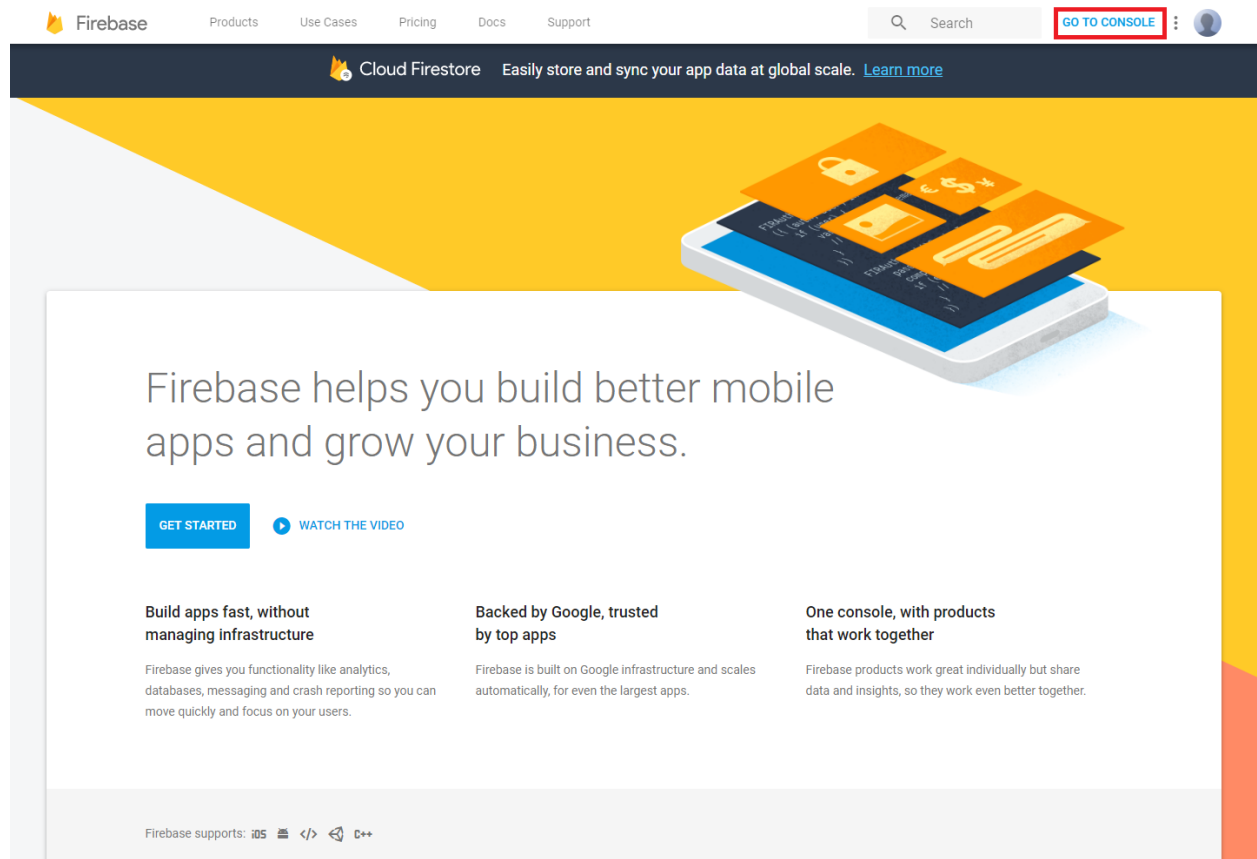
At the bottom, it indicates 'View execution logs in the Firebase console' and 'Last deployed on 12/09/2017 14:57'. A 'DEPLOY' button is visible. On the right, there is a 'Try it now' button and a message: 'Please use test console above to try a sentence.'

#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.



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.firebase.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

Welcome to Firebase!

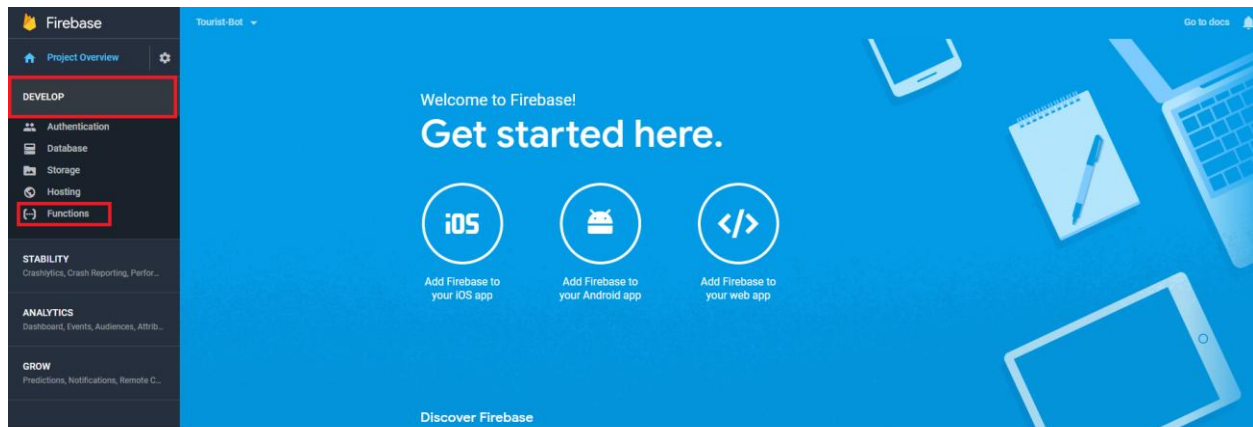
Tools from Google for developing great apps, engaging with your users and earning more through mobile ads.

[Learn more](#) [Documentation](#) [Support](#)

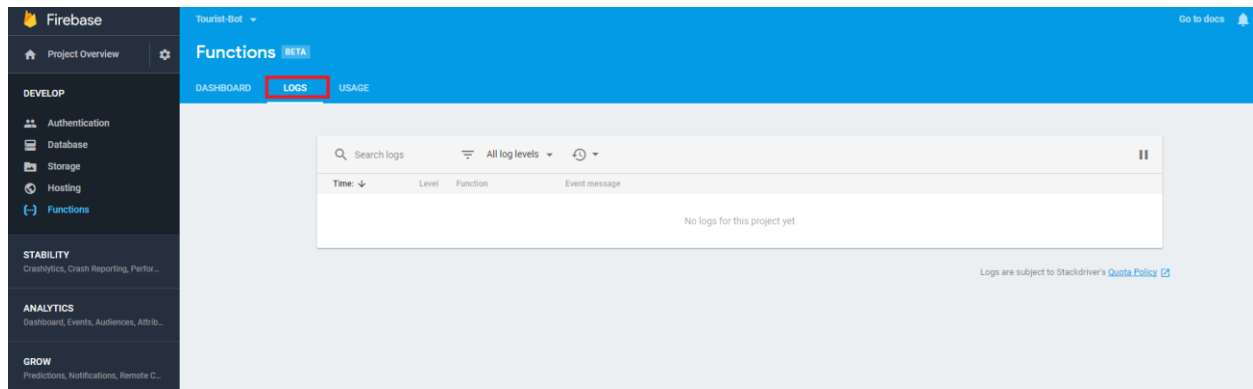
Your projects using Firebase



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

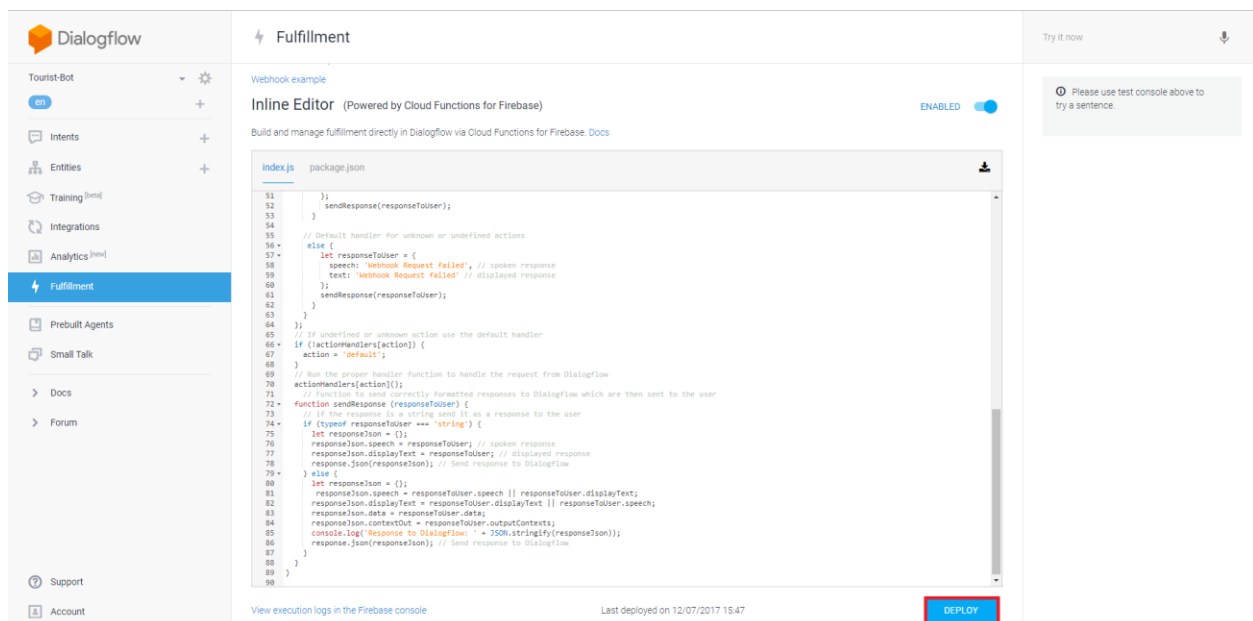


By default, you will get DASHBOARD. You navigate to **LOGS** where you can view logs and usage of functions in Firebase.

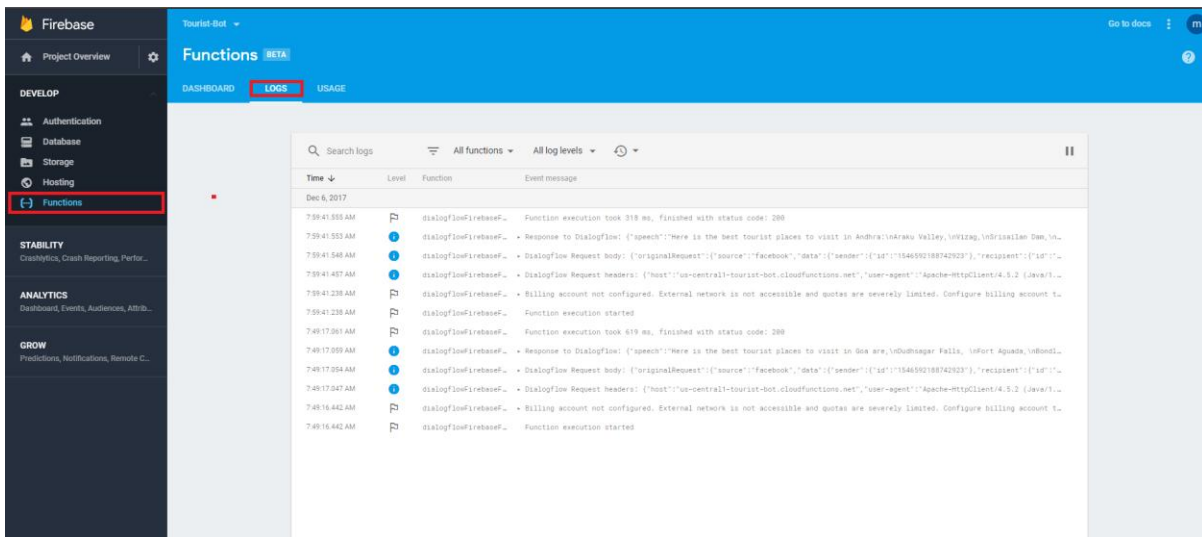


Now click on **DEPLOY** button at the bottom in Dialogflow. Once deployment is done you will get message as **"Successfully deployed your Cloud Function and saved your Webhook"**.

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

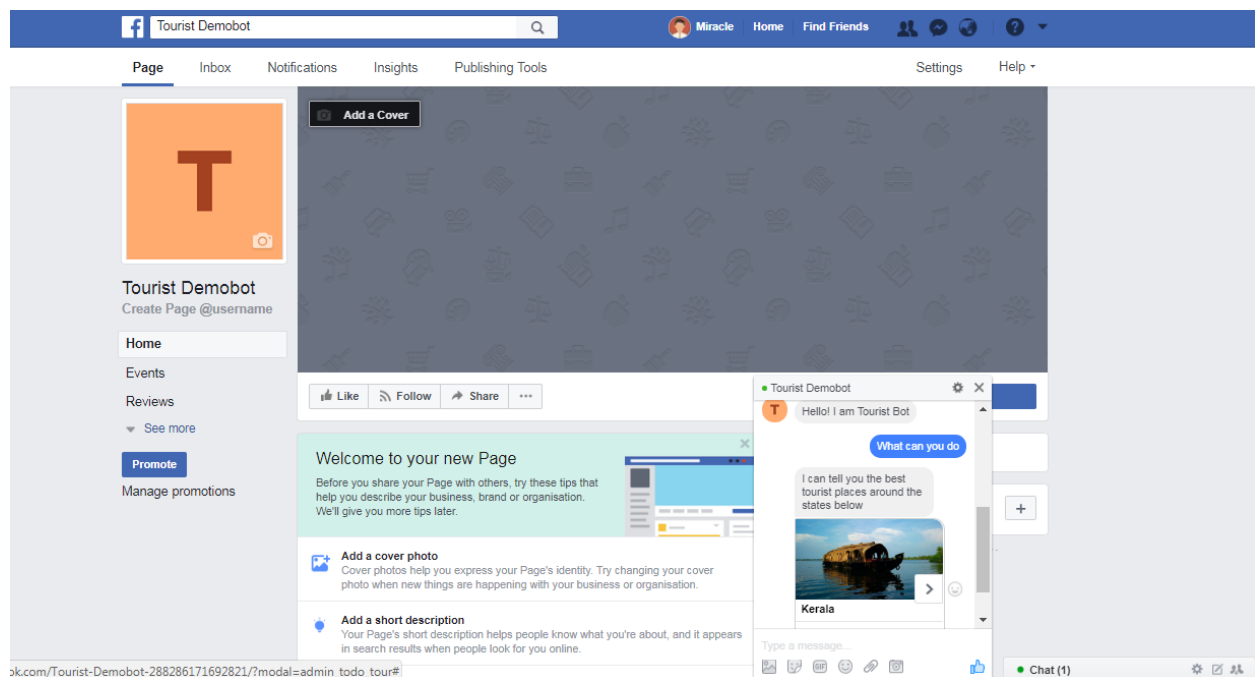


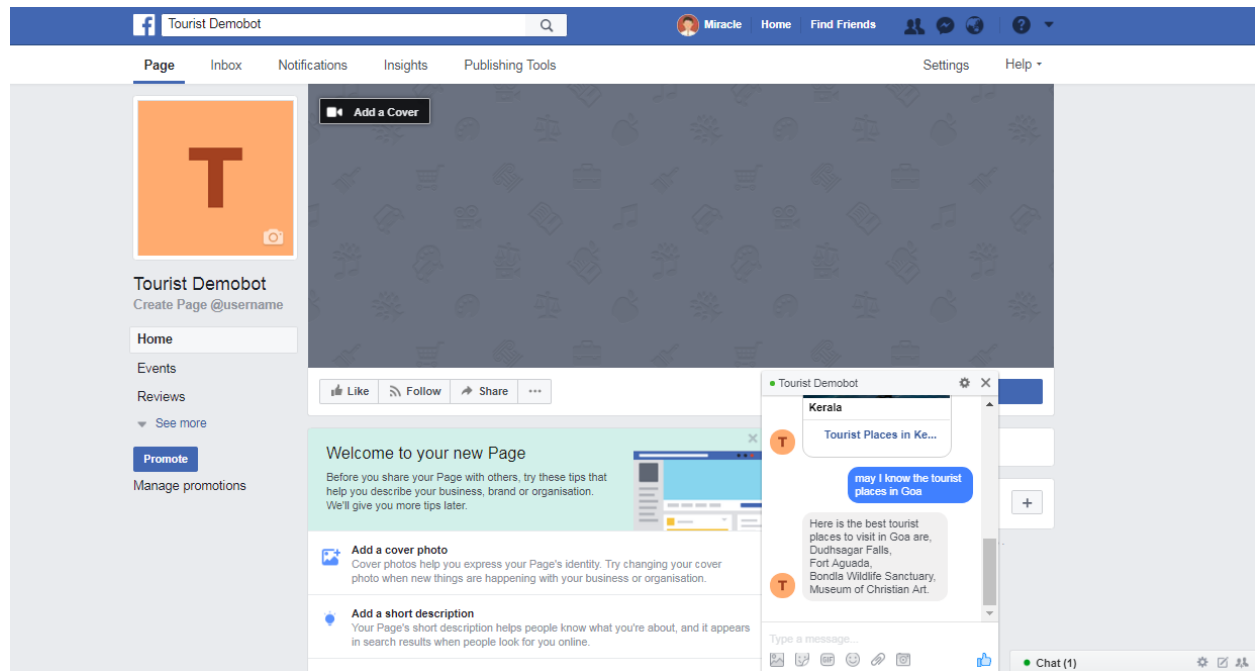
You can check your logs under Functions in Firebase console, once user start interacting with bot <https://console.firebase.google.com/>



Testing Tourist Demo Bot in Facebook Messenger

Open your Facebook Messenger and open the chat window of Tourist DemoBot (It is the 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 😊