Dialogflow Chatbots

Marc Anthony Atanante

Introduction

In this activity, we will develop a Dialogflow chatbot, add it to the Flask Python framework and use Kommunicate to deploy it to our website.

Dialogflow is a Google-owned framework that enables users to develop human-computer interaction technologies that can support Natural Language Processing (NLP) and Flask is a web application framework written in Python.

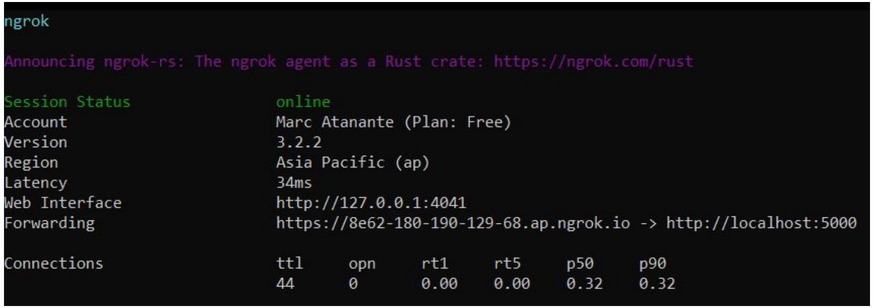
Step 1: Dialogflow and Flask

Go to the Dialogflow website, sign up for an account and follow the steps below:

1. Create an Agent
   - Near the agent settings, click “Create new agent”
   - Provide an agent name (in my case marc-agent) then click create.
2. Create an Intent
   - When an end-user says something, Dialogflow matches the expression to the best intent in your agent.
   - Click the “Create intent” button and provide an intent name (marc-agent-intent).
   - Add training phrases. These are example words/phrases for what end users might say.
3. Enable Fulfillment
   - Enable webhook to help us transfer data and responses between Dialogflow and Flask.
   - The webhook requires a URL (HTTPS protocol). This URL will receive a POST request from Dialogflow each time an intent triggers the webhook.

Step 1: Dialogflow and Flask

4. Setup the Flask Python process by creating a .py file (See Code Snippets).
   - Add a route/webhook method with a POST request. This will execute all the methods.
   - Add a fulfillment text to return a value when a Dialogflow training phrase is detected.
5. Creating a public URL
   - Download, install ngrok and sign up for a free account.
   - Flask is usually using port 5000.
   - Run the command ngrok http 5000 to create a webhook URL that should be added to the Dialogflow Fulfillment.



```
ngrok

Announcing ngrok-rs: The ngrok agent as a Rust crate: https://ngrok.com/rust

Session Status                online
Account                       Marc Atanante (Plan: Free)
Version                       3.2.2
Region                        Asia Pacific (ap)
Latency                       34ms
Web Interface                 http://127.0.0.1:4041
Forwarding                    https://8e62-180-190-129-68.ap.ngrok.io -> http://localhost:5000

Connections                   ttl     opn     rt1     rt5     p50     p90
                              44      0       0.00    0.00    0.32    0.32
```

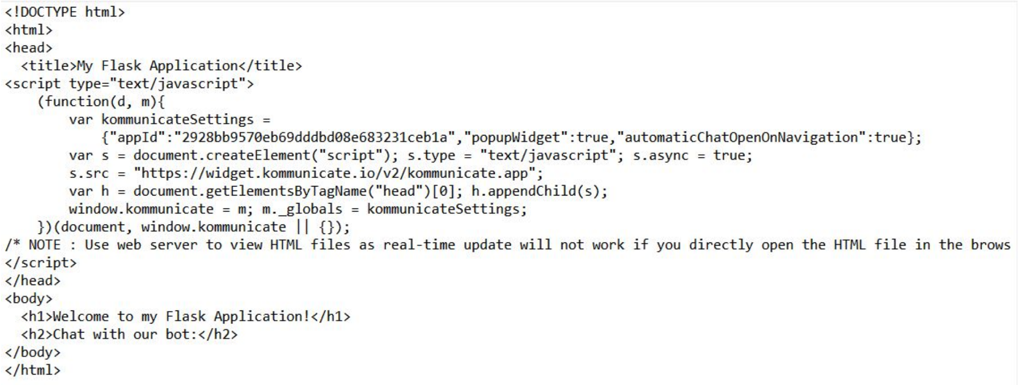
Step 2: Kommunicate

Kommunicate is a customer support automation platform. Sign up for free and follow the steps below:

1. In your Kommunicate dashboard, navigate to “Bot Integration”.
2. Get your Dialogflow API credentials:
   - Open your Dialogflow agent settings and click on the Google cloud link.
   - Go to API’s & Services then go to the Credentials section.
   - Create a service account with “Owner” permission.
   - Go to your service account and create a new JSON key.
3. In the “Bot Integration” section, go to Dialogflow ES.
   - Upload the JSON key
   - Assign a bot name: marc-bot
   - Choose an option to enable/disable bot to human handoff.
   - Go to Conversation Rules and assign new conversations to your bot.

Step 2: Kommunicate

4. Create a folder named “templates” in your working directory and an .HTML file (home.html) inside. This is where you will insert the code for installation.
5. Install the bot to your website.
   ○ Go to Settings -> Install
   ○ Copy the Javascript code inside the <head> section for this to work.

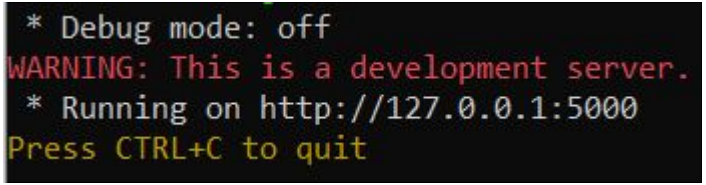


```
<!DOCTYPE html>
<html>
<head>
  <title>My Flask Application</title>
<script type="text/javascript">
    (function(d, m){
        var kommunicateSettings =
            {"appId":"2928bb9570eb69dddbd08e683231ceb1a","popupWidget":true,"automaticChatOpenOnNavigation":true};
        var s = document.createElement("script"); s.type = "text/javascript"; s.async = true;
        s.src = "https://widget.kommunicate.io/v2/kommunicate.app";
        var h = document.getElementsByTagName("head")[0]; h.appendChild(s);
        window.kommunicate = m; m._globals = kommunicateSettings;
    })(document, window.kommunicate || {});
/* NOTE : Use web server to view HTML files as real-time update will not work if you directly open the HTML file in the brows
</script>
</head>
<body>
  <h1>Welcome to my Flask Application!</h1>
  <h2>Chat with our bot:</h2>
</body>
</html>
```

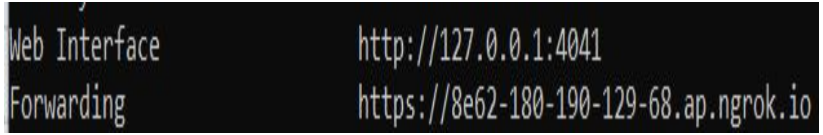
Step 3: Running Flask

With ngrok still running, follow the steps below:

1. Go to your working directory, this should contain:
   - app.py - Code containing the Flask and Dialogflow process.
   - templates - Folder containing your home.html file.
   - JSON file (optional) - Service account private key used to integrate your bot in Kommunicate.
2. Run the commands
   - set FLASK_APP = app
   - flask run
3. Ngrok
   - Go to your ngrok terminal and access the URL in the “Forwarding” section.



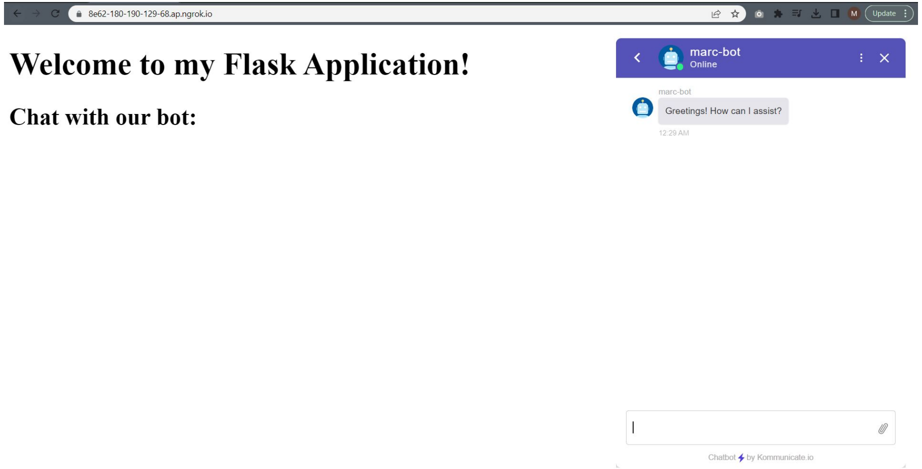
```
 * Debug mode: off
WARNING: This is a development server.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
```



```
Web Interface                 http://127.0.0.1:4041
Forwarding                    https://8e62-180-190-129-68.ap.ngrok.io
```

Step 4: Demonstration

The page should follow the setup of your home.html file. We will then test the chatbot based on our training phrases.



Welcome to my Flask Application!

Chat with our bot:

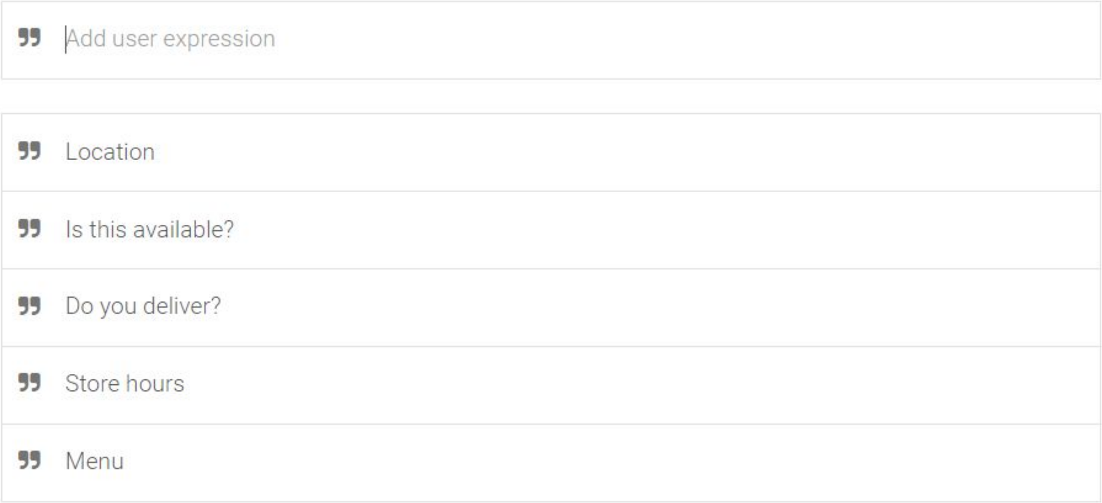
marc-bot

Greetings! How can I assist?

12:29 AM

Step 4: Demonstration

Say that we are deploying a chatbot for a restaurant. These are sample words/phrases that customers mostly use when enquiring:



Add user expression

Location

Is this available?

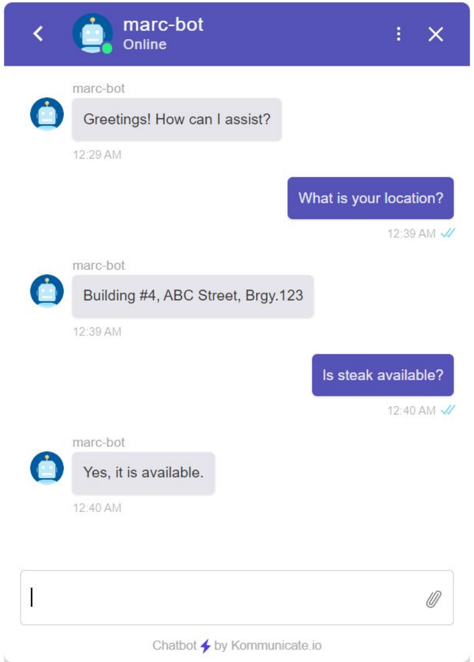
Do you deliver?

Store hours

Menu

Step 4: Demonstration

Talk to the chatbot using expressions that resemble those in the intent training phrases. Depending on your logic in your app.py, this should work smoothly:



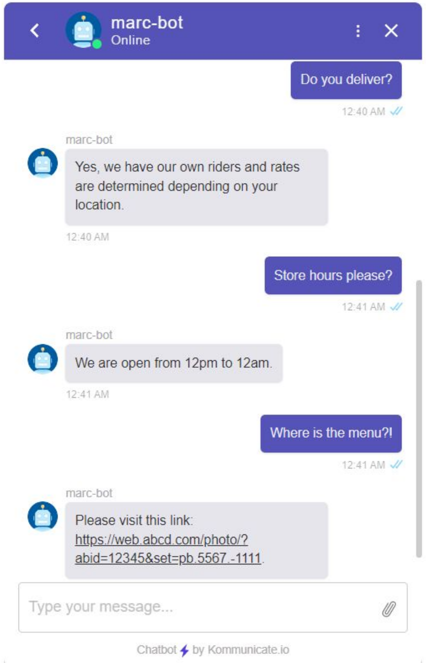
marc-bot
Greetings! How can I assist?
12:29 AM

What is your location?
12:39 AM

marc-bot
Building #4, ABC Street, Brgy.123
12:39 AM

Is steak available?
12:40 AM

marc-bot
Yes, it is available.
12:40 AM



Do you deliver?
12:40 AM

marc-bot
Yes, we have our own riders and rates are determined depending on your location.
12:40 AM

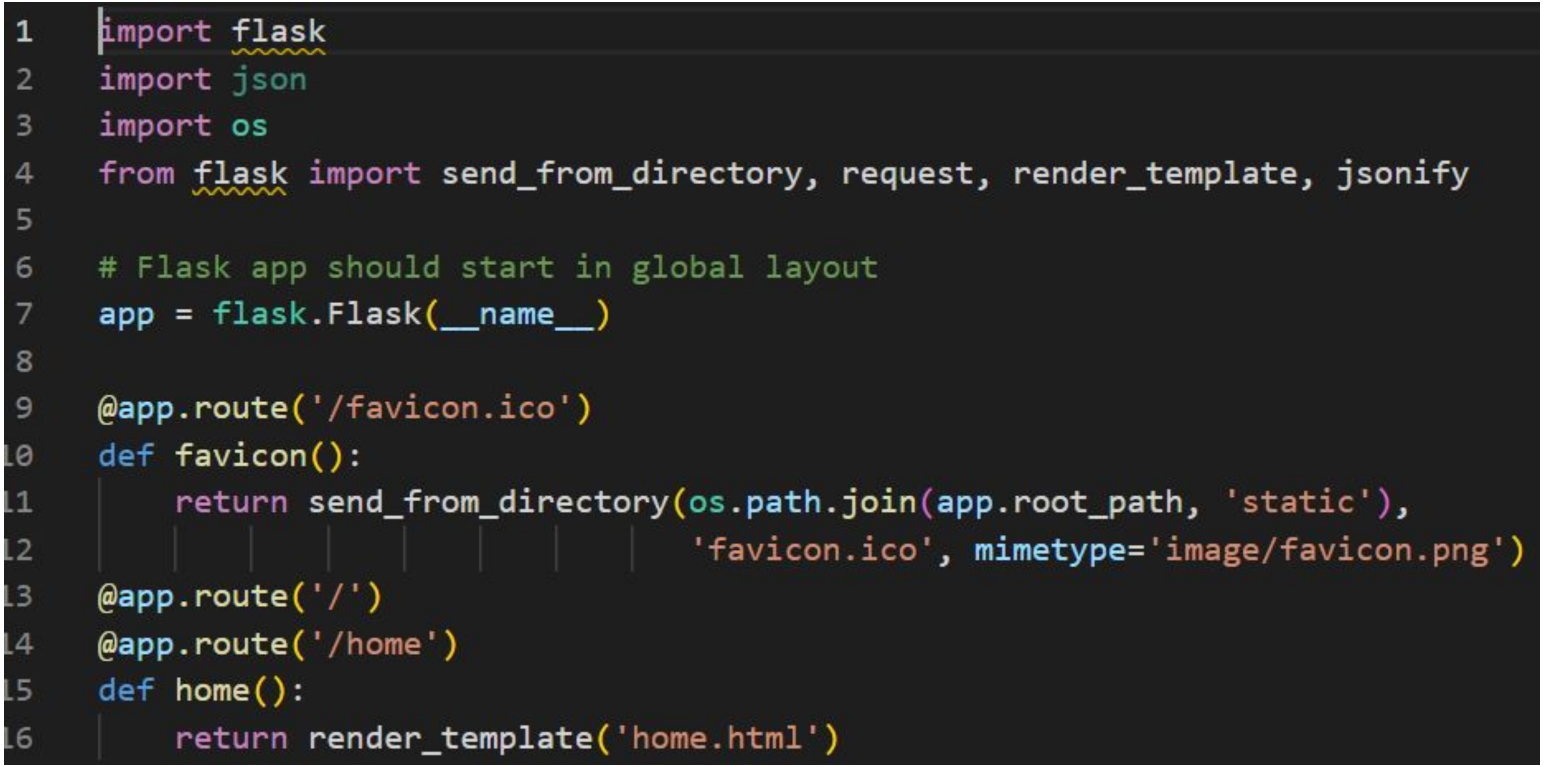
Store hours please?
12:41 AM

marc-bot
We are open from 12pm to 12am.
12:41 AM

Where is the menu?!
12:41 AM

marc-bot
Please visit this link: https://web.abcd.com/photo/?abid=12345&set=pb.5567.-1111.

Code Snippets

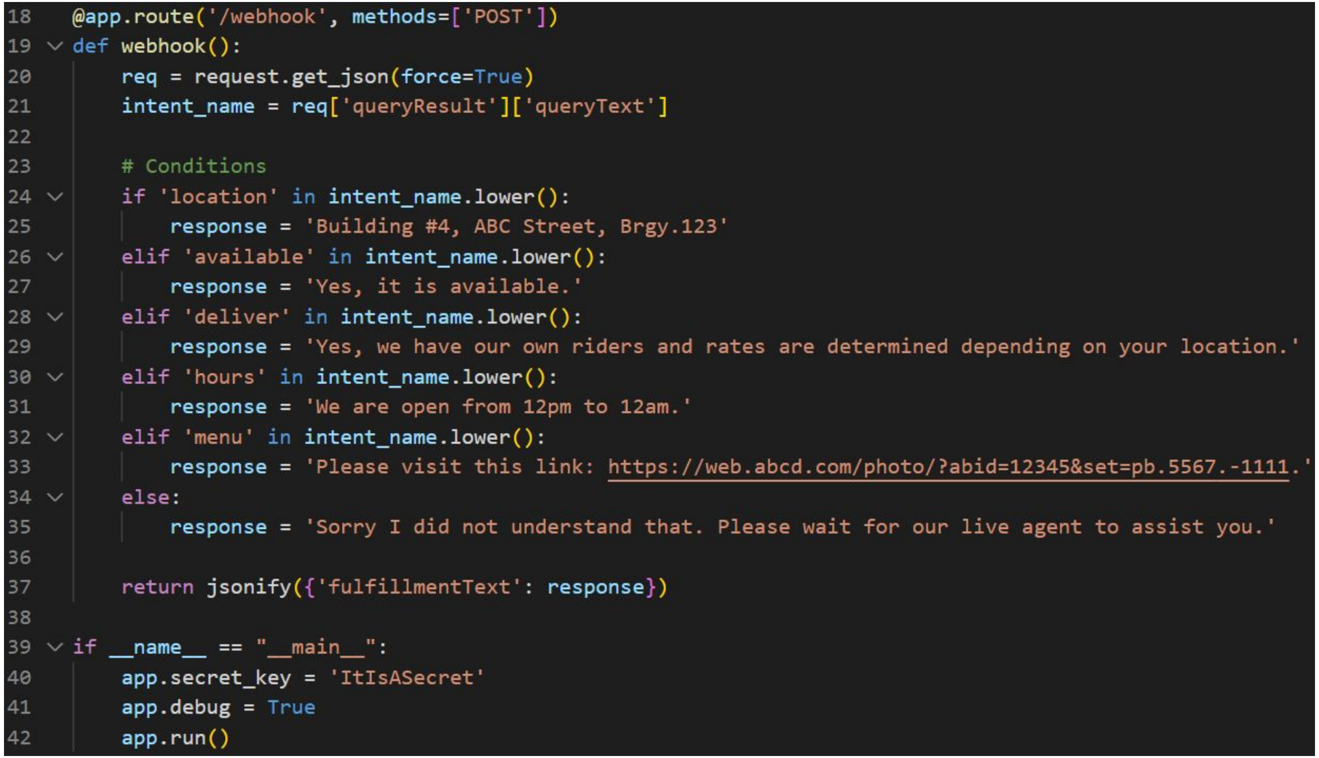


```python
import flask
import json
import os
from flask import send_from_directory, request, render_template, jsonify

# Flask app should start in global layout
app = flask.Flask(__name__)

@app.route('/favicon.ico')
def favicon():
    return send_from_directory(os.path.join(app.root_path, 'static'),
                               'favicon.ico', mimetype='image/favicon.png')
@app.route('/')
@app.route('/home')
def home():
    return render_template('home.html')
```

Code Snippets



```python
@app.route('/webhook', methods=['POST'])
def webhook():
    req = request.get_json(force=True)
    intent_name = req['queryResult']['queryText']

    # Conditions
    if 'location' in intent_name.lower():
        response = 'Building #4, ABC Street, Brgy.123'
    elif 'available' in intent_name.lower():
        response = 'Yes, it is available.'
    elif 'deliver' in intent_name.lower():
        response = 'Yes, we have our own riders and rates are determined depending on your location.'
    elif 'hours' in intent_name.lower():
        response = 'We are open from 12pm to 12am.'
    elif 'menu' in intent_name.lower():
        response = 'Please visit this link: https://web.abcd.com/photo/?abid=12345&set=pb.5567.-1111.'
    else:
        response = 'Sorry I did not understand that. Please wait for our live agent to assist you.'

    return jsonify({'fulfillmentText': response})

if __name__ == "__main__":
    app.secret_key = 'ItIsASecret'
    app.debug = True
    app.run()
```