Whatsapp chatbot with Python and Twilio

Date: 2023-11-15
By: csr13

Download as PDF

Chatbots are everywhere, but sometimes a more custom chatbot can be built for your business needs, more so if your business takes orders, books rooms, sends reminders, etc. Many services already do this, but you can't customize them, they are generic, and cost a lot.

They key, is make bot as cheap as possible with little as possible, cut down on operational costs, this makes programmable whatsapp bots good. Even better if you have an api that your business depends on with endpoints to check your database of products for availability, rooms, promotions, etc.

Enough chit chat, let's get to the code.

You need.

- Twilio account.
- Twilio phone number (us numbers free)
- Some place to host the code, as it needs to be exposed via http server, as twilio acts as a middleman message broker.

Twilio number cost is very cheap once free trial is over, perhaps 5 usd per month on a very very busy bot, add an ec2 instance (hosting code) maybe 3.00 usd per month, or custom on prem machine, let's say tops $ 8.00 usd per month per bot that is very busy.

Here is the code for a simple chatbot, it's written in python.

```
import os
from flask import Flask, request, jsonify

from twilio.twiml.messaging_response import MessagingResponse
from twilio.rest import Client

# You need to get these from twilio, and you need to
# register this url on twilio sandbox settings.

TWILIO_ACCOUNT_SID = os.getenv(
    "TWILIO_ACCOUNT_SID",
    "twilio account sid"
)
TWILIO_AUTH_TOKEN = os.getenv(
    "TWILIO_AUTH_TOKEN",
    "Twilio auth token"
)
TWILIO_TEST_AUTH_TOKEN = os.getenv(
    "TWILIO_TEST_AUTH_TOKEN",
    "Twilio test auth token"
)


app = Flask(__name__)
client = Client(
    TWILIO_ACCOUNT_SID,
    TWILIO_TEST_AUTH_TOKEN
)

greeting = """
Hola 😀

1) Para disponibilidad y horarios.
2) Para nuestros horarios.
3) Para ver este menu.
"""

menu = """

MENU:

Ingresa el numbero de opcion que quieras.

1) Para disponibilidad y horarios.
2) Para nuestros horarios.
3) Para ver este menu.
"""

def respond(message):
```

```
    response = MessagingResponse()
    response.message(message)
    return str(response)

@app.route("/message", methods=("POST",))
def message():
    """
    THis is the webook twilio will send messages to, and it will respond according to
    user input options, the response is actually xml response to twilio.
    """
    message = request.form.get("Body")
    if message is None:
        return jsonify(error="Missing message"), 400
    message = message.lower()
    try:
        # Handle the options that the customer selected.
        # a more elaborate bot, has submenus of main menus
        # and so on, this is for example purposes.
        if message == '1':
            return respond(
                "Visitanos aqui para reservar \n\n www.tu sitio.com"
            )
        elif message == "2":
            return respond("12:00pm - 6:00pm estamos disponibles.")
        elif message == "3":
            return respond(menu)
        else:
            return respond(greeting)
    except Exception as error:
        return respond(str(error))
    return  respond(greeting)


if __name__ == "__main__":
    app.run(debug=True, host="127.0.0.1", port=8888)
```

You could add another option that makes a request to your custom bookings API server, to check for rooms available, send options, and redirect them with a link to your website so they can book that specific room if they decide they want it, or something else.

The message webhook would look like this if option 3 would send back availabilities, for tables, rooms, or seats (for example)

```
@app.route("/message", methods=("POST",))
def message():
    """
    THis is the webook twilio will send messages to, and it will respond according to
    user input options, the response is actually xml response to twilio.
    """
    message = request.form.get("Body")
    if message is None:
        return jsonify(error="Missing message"), 400
    message = message.lower()
    try:
        # Handle the options that the customer selected.
        # a more elaborate bot, has submenus of main menus
        # and so on, this is for example purposes.
        if message == '1':
            return respond(
                "Visitanos aqui para reservar \n\n www.tu sitio.com"
            )
        elif message == "2":
            return respond("12:00pm - 6:00pm estamos disponibles.")
        elif message == "3":
            try:
                resp = requests.get(
                    "http://my-biz/api/v1/availabilities",
                    headers={"API_KEY" : "<auth key>"}
                )
                if resp.status_code != 200:
                    return respond("No puedo conseguir las disponibilidades.")
                else:
                    return respond(resp.json()["disponibilidades_con_links"])
        elif message == "4":
            return respond(menu)
        else:
            return respond(greeting)
    except Exception as error:
        return respond(str(error))
    return  respond(greeting)
```

It is up to your client what he/she wants make available through their whatsapp bot, with Twilio the posibilities to build interactive chatbots are very large and one can get very creative.

Contact me if this is something you would want for you or your business.

Related Notes

1) Using ec2 instances as sneaker bid bots pt 2.
Download PDF
Date published: 2023-11-27
bots python aws

2) Using ec2 instances as sneaker bid bots pt 1.
Download PDF
Date published: 2023-11-21
bots python aws

3) Real Time Language Translation Agent System for Call Centers
Download PDF
Date published: 2023-11-16
voip telephony python systems

4) Using ec2 instances as sneaker bid bots pt 2.
Download PDF
Date published: 2023-11-27
bots python aws

5) Backend Celery task manager dashboard via Flower
Download PDF
Date published: 2023-10-29
backend tasks python

6) Using ec2 instances as sneaker bid bots pt 1.
Download PDF
Date published: 2023-11-21
bots python aws