

# WHATSAPP BOT

## Requirements:

1. Python environment
2. Download all necessary package – flask, requests, twilio
3. Ngrok – download it (Used for converting the flask to an URL which communicates with twilio)

## Twilio Configuration:

Signup with a free account in [twilio.com](https://www.twilio.com) and add your phone number which has whatsapp installed.

Twilio provides a WhatsApp sandbox where you can easily develop and test your application. Once your application is complete you can request production access for your Twilio phone number, which requires approval by WhatsApp.

Let's connect your smartphone to the sandbox. From your Twilio Console, select Programmable Messaging, then click on "Try it Out" and finally click on Try WhatsApp. The WhatsApp sandbox page will show you the sandbox number assigned to your account, and a join code.

Or go to this [link](#) directly. Create an account on twilio.

## STEP 1:

### Pip install Twilio flask Requests

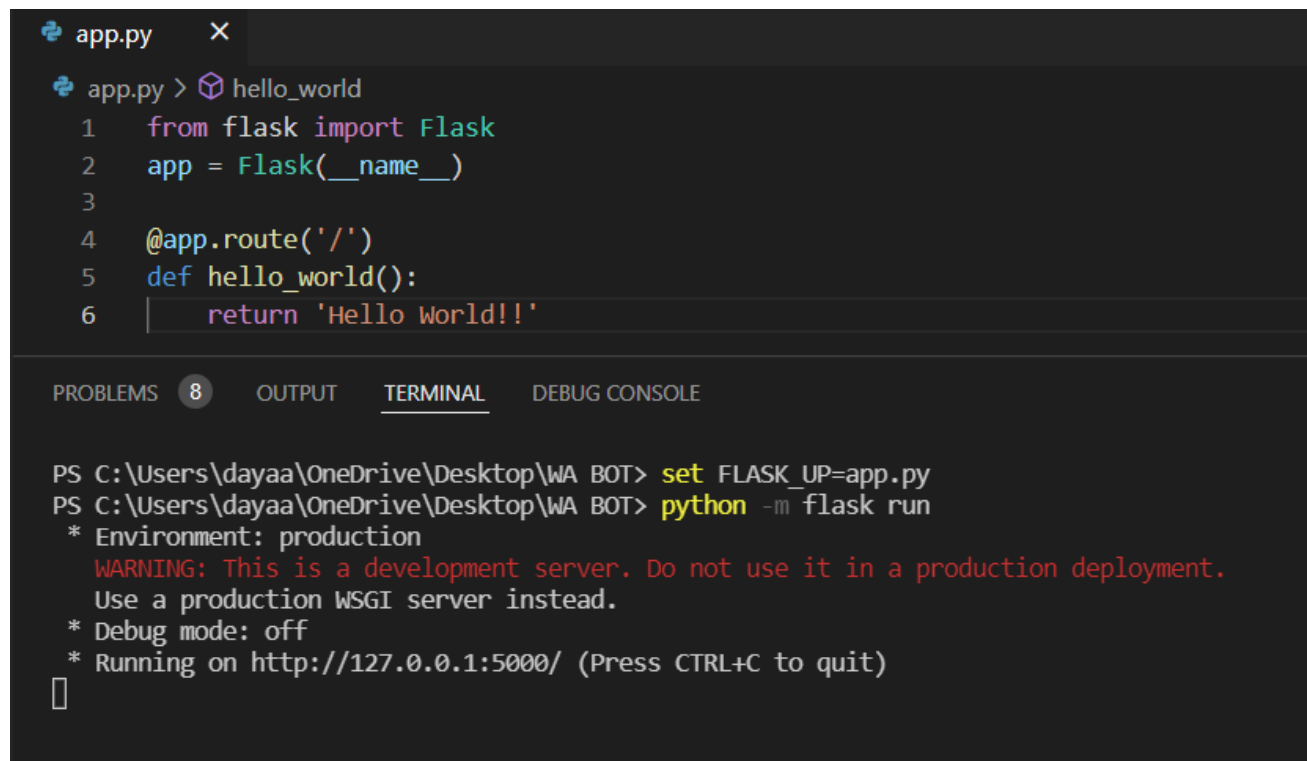
```
PS C:\Users\dayaa\OneDrive\Desktop\WA BOT> pip install Twilio flask Requests
Requirement already satisfied: Twilio in c:\users\dayaa\appdata\local\programs\python\python39\lib\site-packages (7.3.2)
Requirement already satisfied: flask in c:\users\dayaa\appdata\local\programs\python\python39\lib\site-packages (2.0.1)
Requirement already satisfied: Requests in c:\users\dayaa\appdata\local\programs\python\python39\lib\site-packages (2.26.0)
Requirement already satisfied: pytz in c:\users\dayaa\appdata\local\programs\python\python39\lib\site-packages (from Twilio) (2021.1)
Requirement already satisfied: PyJWT<3.0.0,>=2.0.0 in c:\users\dayaa\appdata\local\programs\python\python39\lib\site-packages (from Twilio) (2.3.0)
Requirement already satisfied: click>=7.1.2 in c:\users\dayaa\appdata\local\programs\python\python39\lib\site-packages (from flask) (8.0.1)
Requirement already satisfied: Werkzeug>=2.0 in c:\users\dayaa\appdata\local\programs\python\python39\lib\site-packages (from flask) (2.0.1)
Requirement already satisfied: itsdangerous>=2.0 in c:\users\dayaa\appdata\local\programs\python\python39\lib\site-packages (from flask) (2.0.1)
Requirement already satisfied: Jinja2>=3.0 in c:\users\dayaa\appdata\local\programs\python\python39\lib\site-packages (from flask) (3.0.1)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\dayaa\appdata\local\programs\python\python39\lib\site-packages (from Requests) (1.26.6)
Requirement already satisfied: idna<4,>=2.5 in c:\users\dayaa\appdata\local\programs\python\python39\lib\site-packages (from Requests) (3.2)
Requirement already satisfied: charset-normalizer<=2.0.0 in c:\users\dayaa\appdata\local\programs\python\python39\lib\site-packages (from Requests) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dayaa\appdata\local\programs\python\python39\lib\site-packages (from Requests) (2021.5.30)
Requirement already satisfied: colorama in c:\users\dayaa\appdata\local\programs\python\python39\lib\site-packages (from click>=7.1.2->flask) (0.4.4)
Requirement already satisfied: MarkupSafe>=2.0 in c:\users\dayaa\appdata\local\programs\python\python39\lib\site-packages (from Jinja2>=3.0->flask) (2.0.1)
WARNING: You are using pip version 21.3.1; however, version 22.0.3 is available.
You should consider upgrading via the 'c:\users\dayaa\appdata\local\programs\python\python39\python.exe -m pip install --upgrade pip' command.
PS C:\Users\dayaa\OneDrive\Desktop\WA BOT> []
```

- ❖ The Flask framework, to create the web application
- ❖ The Twilio Python Helper library, to work with the Twilio APIs
- ❖ The Requests package, to access third party APIs

## STEP 2:

Run a sample program to check all the accessories is working perfectly. Type the code save app.py

Run it by:



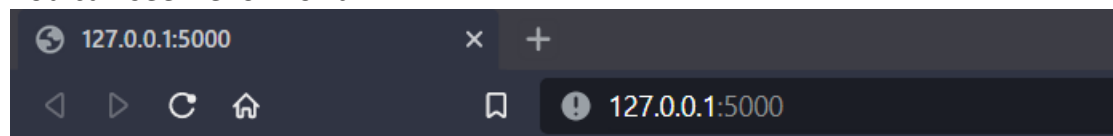
```
app.py x
app.py > hello_world
1  from flask import Flask
2  app = Flask(__name__)
3
4  @app.route('/')
5  def hello_world():
6      return 'Hello World!!'
```

PROBLEMS 8 OUTPUT TERMINAL DEBUG CONSOLE

```
PS C:\Users\dayaa\OneDrive\Desktop\WA BOT> set FLASK_UP=app.py
PS C:\Users\dayaa\OneDrive\Desktop\WA BOT> python -m flask run
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Type the URL given in any web browser (preferably chrome) or press ctrl+enter in windows machine to open up the link when hovering the link in Visual Studio Code.

You can see Hello World



Hello World!!

### STEP 3:

Start writing the code for bot creation (blueprint refer any website / my app.py)

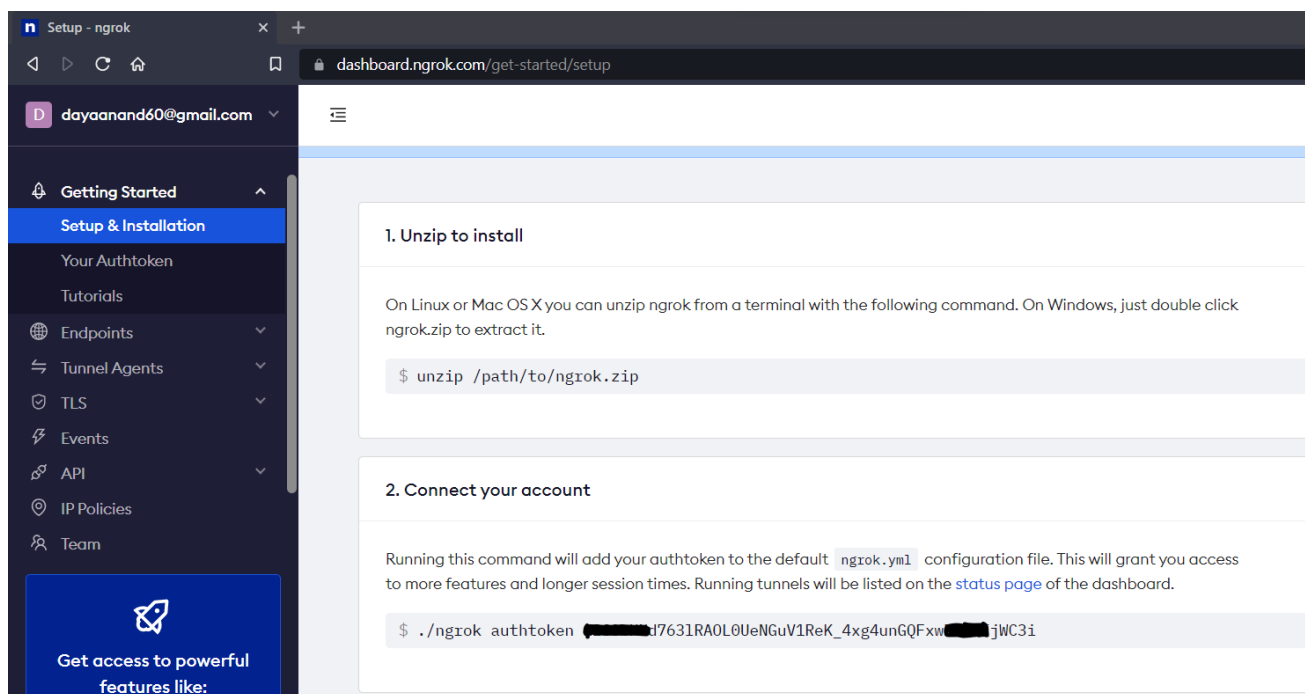
```
from flask import Flask
app = Flask(__name__)
```

```
@app.route('/')
def hello_world():
    return 'Hello World!!'
```

### STEP 4:

Installation of ngrok to convert the flask app to an URL which connects to twilio. Download ngrok from <https://ngrok.com/download> and unzip and run the .exe file.

Signup in ngrok website where you can get authorized token in ngrok.



In command prompt (open that ngrok.exe file)

```
C:\Users\dayaa\OneDrive\Desktop\WA BOT\ngrok-stable-windows-amd64>ngrok authtoken [token]
Auth token saved to configuration file: C:\Users\dayaa\.ngrok2\ngrok.yml
C:\Users\dayaa\OneDrive\Desktop\WA BOT\ngrok-stable-windows-amd64>
```

## STEP 5:

Now run the python flask code:

Open new command prompt run ngrok http 5000 to allocate a temporary public domain that redirects HTTP requests to our local port 5000.

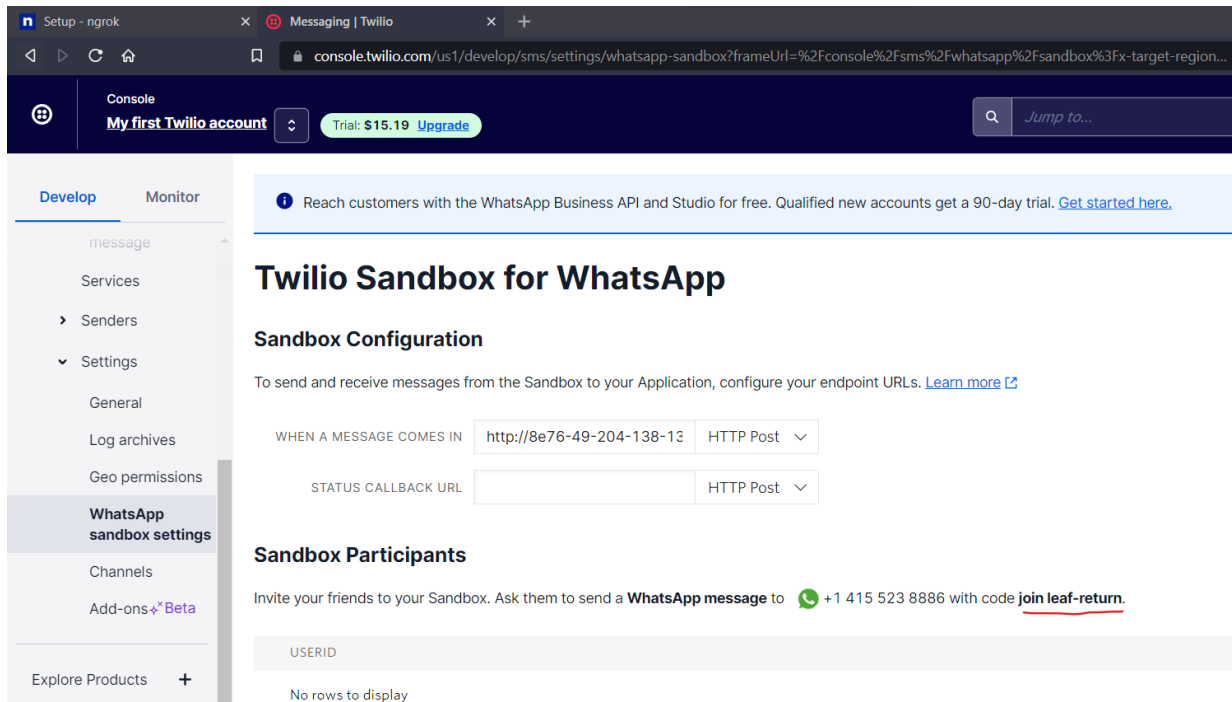
```
ngrok by @inconshreveable

Session Status      online
Account             dayaanand60@gmail.com (Plan: Free)
Version             2.3.40
Region             United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           http://8e76-49-204-138-135.ngrok.io -> http://localhost:5000
Forwarding           https://8e76-49-204-138-135.ngrok.io -> http://localhost:5000

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

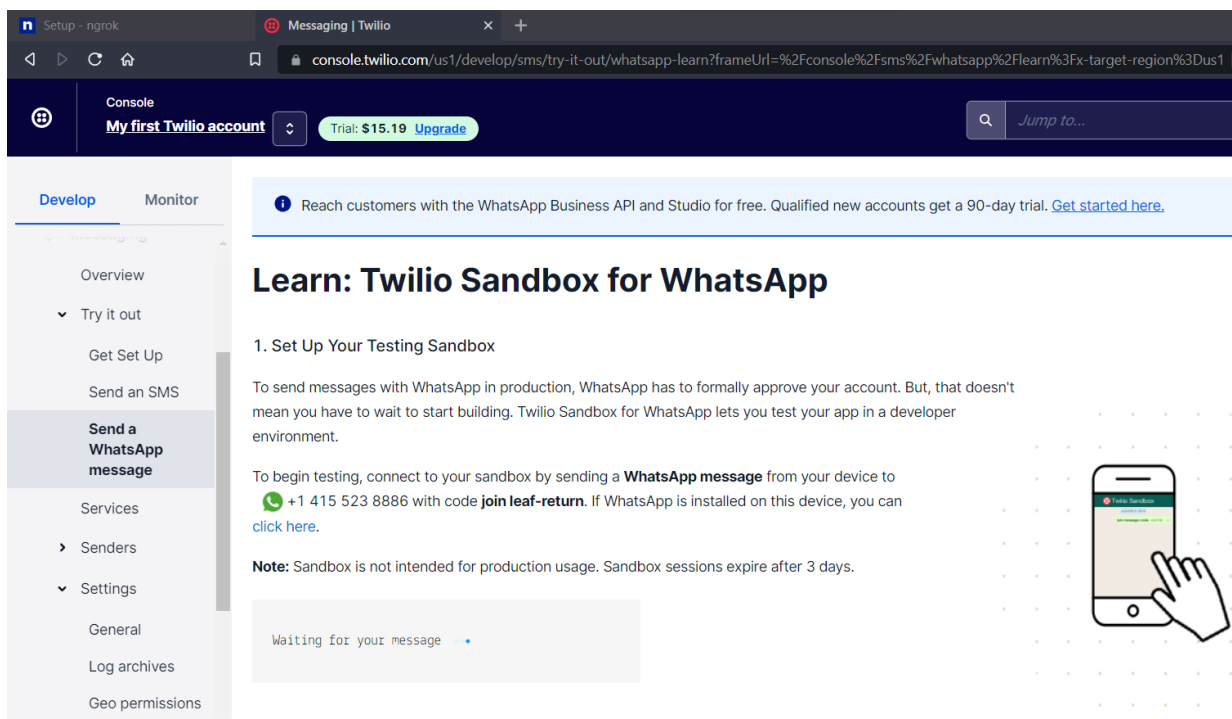
Note the lines beginning with “Forwarding”. These show the public URL that ngrok uses to redirect requests into our service. What we need to do now is tell Twilio to use this URL to send incoming message notifications.

Go back to the Twilio Console, click on Programmable Messaging, then on Settings, and finally on WhatsApp Sandbox Settings. Copy the http:// URL from the ngrok output and then paste it on the “When a message comes in” field. Since our chatbot is exposed under the **/bot** URL, append that at the end of the root ngrok URL. Make sure the request method is set to HTTP Post. Don’t forget to click the red Save button at the bottom of the page to record these changes.



The screenshot shows the Twilio Console interface. The left sidebar has a menu with 'Develop' and 'Monitor' tabs. Under 'Develop', 'WhatsApp sandbox settings' is selected. The main content area is titled 'Twilio Sandbox for WhatsApp' and 'Sandbox Configuration'. It includes a message: 'Reach customers with the WhatsApp Business API and Studio for free. Qualified new accounts get a 90-day trial. [Get started here.](#)' Below this, there's a section for 'Sandbox Configuration' with fields for 'WHEN A MESSAGE COMES IN' (set to 'http://8e76-49-204-138-13' and 'HTTP Post') and 'STATUS CALLBACK URL' (set to 'HTTP Post'). A 'Sandbox Participants' section follows, with an invite link: '+1 415 523 8886 with code join leaf-return'. At the bottom, there's a 'USERID' field and a message 'No rows to display'.

**Add /bot to the end of the http link - Important**



The screenshot shows the Twilio Console interface. The left sidebar has a menu with 'Develop' and 'Monitor' tabs. Under 'Develop', 'Send a WhatsApp message' is selected. The main content area is titled 'Learn: Twilio Sandbox for WhatsApp' and '1. Set Up Your Testing Sandbox'. It includes a message: 'To send messages with WhatsApp in production, WhatsApp has to formally approve your account. But, that doesn't mean you have to wait to start building. Twilio Sandbox for WhatsApp lets you test your app in a developer environment.' Below this, there's a section for 'Send a WhatsApp message' with a message: 'To begin testing, connect to your sandbox by sending a **WhatsApp message** from your device to +1 415 523 8886 with code join leaf-return. If WhatsApp is installed on this device, you can [click here](#).' A 'Note' states: 'Sandbox is not intended for production usage. Sandbox sessions expire after 3 days.' At the bottom, there's a 'Waiting for your message' status. On the right, there's an illustration of a hand sending a message to a smartphone.

Before setup the sandbox in twilio – connects your number to twilio. Send a join leaf-return to connect.

Then as per your code, the twilio works

```
C:\Users\dayaa\OneDrive\Desktop\WA BOT\ngrok-stable-windows-amd64\ngrok.exe - ngrok http 5000
ngrok by @inconshreveable

Session Status      online
Account             dayaanand60@gmail.com (Plan: Free)
Version             2.3.40
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           http://8e76-49-204-138-135.ngrok.io -> http://localhost:5000
Forwarding           https://8e76-49-204-138-135.ngrok.io -> http://localhost:5000

Connections          ttl      opn      rt1      rt5      p50      p90
                   6        0        0.02     0.01     0.32     0.33

HTTP Requests
-----
POST /bot            200 OK
POST /bot            200 OK
POST /bot            200 OK
```

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\dayaa\OneDrive\Desktop\WA BOT> set FLASK_UP=app.py
PS C:\Users\dayaa\OneDrive\Desktop\WA BOT> python -m flask run
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
127.0.0.1 - - [09/Feb/2022 00:14:13] "POST /bot HTTP/1.1" 200 -
127.0.0.1 - - [09/Feb/2022 00:14:26] "POST /bot HTTP/1.1" 200 -
127.0.0.1 - - [09/Feb/2022 00:14:49] "POST /bot HTTP/1.1" 200 -
█
```

```

C:\Users\dayaa\OneDrive\Desktop\WA BOT\ngrok-stable-windows-amd64\ngrok.exe - ngrok http 5000
ngrok by @inconshreveable

Session Status      online
Account             dayaanand60@gmail.com (Plan: Free)
Version             2.3.40
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://8e76-49-204-138-135.ngrok.io -> http://localhost:5000
Forwarding           https://8e76-49-204-138-135.ngrok.io -> http://localhost:5000

Connections          ttl      opn      rt1      rt5      p50      p90
                    9        0        0.02     0.02     0.33     2.38

HTTP Requests
-----
POST /bot            200 OK
POST /bot            200 OK
POST /bot            200 OK
POST /bot            200 OK
POST /bot            200 OK
POST /bot            200 OK
POST /bot            200 OK
POST /bot            200 OK
POST /bot            200 OK
POST /bot            200 OK

```

Now you can start sending messages to the chatbot from the smartphone that you connected to the sandbox. You can type any sentences that you like, and each time the words “cat” appear in messages the chatbot will invoke the third-party APIs and return some fresh content to you.

Keep in mind that when using ngrok for free there are some limitations. In particular, you cannot hold on to a ngrok URL for more than 8 hours, and the domain name that is assigned to you will be different every time you start the ngrok command. You will need to update the URL in the Twilio Console every time you restart ngrok.



Screenshot to fetch the picture of the cat with a quote and to fetch the YouTube link:

