

Voicemail Recording

Project Report

Submitted By:

Kriti Gupta

kriti.gupta@sjsu.edu

<https://www.linkedin.com/in/kriti-gupta-sjsu/>

Table of Contents

TABLE OF CONTENTS	I
ACKNOWLEDGEMENT	ERROR! BOOKMARK NOT DEFINED.
INTRODUCTION	1
PROBLEM STATEMENT:	1
REQUIREMENTS:	1
PROJECT SETUP	2
RESULT	5
REFERENCES	6

Introduction

Problem Statement:

In this project, we need to develop a code to record a voicemail and download it in mp3 format, which would then be uploaded to an AWS S3 bucket. This would allow us to generate a link for that mp3 file which can further be used by anyone to download that mp3 file into their system. Following are the steps involved in this project:

Step 1: Sign up for a number through Twilio.com and record a voicemail with it.

Step 2: Write a program to download the voicemail in mp3 format and then upload it to the Amazon S3 bucket.

Step 3: Share the URL of the voicemail in the S3 bucket as a result.

Requirements:

For this project we first set up a virtual environment. So, install virtualenv:

```
cd desktop/my_quickstart_folder  
sudo pip install virtualenv
```

Then, navigate to my_quickstart_folder and create a virtual environment:

```
virtualenv --system-site-packages .
```

Now install the following requirements for setting up this project:

- Python 2.7 or above
- Twilio 6.0.0 or above
- Flask 0.12 or above
- Boto3 1.16.11 or above
- Twilio account SID (Automatically generated after signing up in Twilio.com)
- Twilio authentication token (Automatically generated after signing up in Twilio.com)
- AWS access key id (Automatically generated after signing up in AWS)
- AWS access key secret (Automatically generated after signing up in AWS)
- AWS S3 bucket name

Once Twilio account SID, Twilio authentication token, AWS access key id, AWS access key secret and AWS S3 bucket name are obtained update them in credentials.py present in the my_quickstart_folder folder.

Project Setup

After setting up the requirements the following steps will show how to get the Python script running:

1. In the terminal or command prompt, navigate to the path to my_quickstart_folder:
`cd desktop/my_quickstart_folder`
2. Now activate the virtual environment
`source bin/activate`
3. Then run the python script voicemail-project.py with the command:
`python voicemail-project.py`

Once the script has started the terminal should look like this:

```
Last login: Wed Nov  4 18:03:24 on ttys000
kriti@Kritis-MacBook-Air ~ % cd desktop/my_quickstart_folder
(my_quickstart_folder) kriti@Kritis-MacBook-Air my_quickstart_folder % python voicemail-project.py
/Users/kriti/Desktop/my_quickstart_folder/lib/python2.7/site-packages/OpenSSL/crypto.py:12: CryptographyDeprecation
Warning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography,
and will be removed in a future release.
  from cryptography import x509
* Serving Flask app "voicemail-project" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
/Users/kriti/Desktop/my_quickstart_folder/lib/python2.7/site-packages/OpenSSL/crypto.py:12: CryptographyDeprecation
Warning: Python 2 is no longer supported by the Python core team. Support for it is now deprecated in cryptography,
and will be removed in a future release.
  from cryptography import x509
* Debugger is active!
* Debugger PIN: 269-431-041
127.0.0.1 - - [04/Nov/2020 18:04:53] "GET / HTTP/1.1" 404 -
127.0.0.1 - - [04/Nov/2020 18:04:54] "GET /favicon.ico HTTP/1.1" 404 -
```

4. Start another terminal and navigate to the path where ngrok driver (Flask server) is installed
`cd desktop/ my_quickstart_folder`
5. Start the flask server with the following command:
`./ngrok http 5000`

Once the server has started the terminal should look as follows:

```
ngrok by @inconshreveable (Ctrl+C to quit)

Session Status      online
Session Expires     7 hours, 50 minutes
Version             2.3.35
Region              United States (us)
Web Interface        http://127.0.0.1:4040
Forwarding           http://e1ac42dc1b09.ngrok.io -> http://localhost:5000
                    https://e1ac42dc1b09.ngrok.io -> http://localhost:5000

Connections         ttl    opn    rt1    rt5    p50    p90
                   2      0      0.00   0.00   0.01   0.01

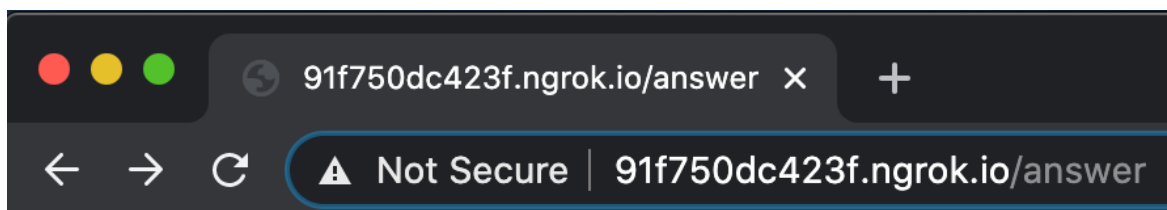
HTTP Requests
-----
GET /favicon.ico    404 NOT FOUND
GET /               404 NOT FOUND
```

Copy the highlighted part, it represents the localhost url where the Twilio number will become live.

6. Go to Twilio.com and login into your account. In the side menu go to 'All products and services' and select the option 'phone numbers' in it. Then click on the number that you want to use for this project. This will open up configurations for that phone number.
7. In the Voice & Fax section, under 'A call comes in' option, select webhook and update the Flask URL by appending /answer to the URL we copied in step 5. Then, click on Save. The following image explains this:

The screenshot shows the Twilio console interface for configuring a phone number. On the left is a sidebar with navigation links: Home, Phone Numbers, Manage Numbers, Active Numbers, Released Numbers, Buy a Number, Verified Caller IDs, Port & Host, Regulatory Compliance, Tools, Usage, and Getting Started. The main content area is titled 'Voice & Fax' and includes a success message: 'Number was successfully updated.' Below this, there are several configuration sections: 'ACCEPT INCOMING' (set to 'Voice Calls'), 'CONFIGURE WITH' (set to 'Webhooks, TwiML Bins, Functions, Studio, or Proxy'), 'A CALL COMES IN' (set to 'Webhook' with URL 'http://91f750dc423f.ngrok.io/answer' and method 'HTTP POST'), 'PRIMARY HANDLER FAILS' (set to 'Webhook' with method 'HTTP POST'), 'CALL STATUS CHANGES' (set to 'HTTP POST'), 'CALLER NAME LOOKUP' (set to 'Disabled'), and 'EMERGENCY CALLING'.

8. Once this step is done, it means that the number is live. We can visit the Flask URL and see that the following webpage should be displayed.



Please leave a message after the beep. Press star when finished

9. From there we can also navigate to /end_call page by replacing answer with end_call in the URL to see that it is working correctly and the following webpage should be displayed.

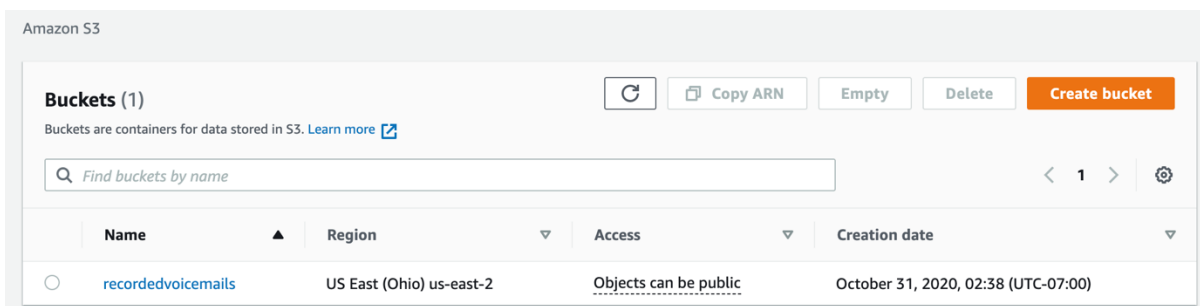


Thank you for calling. Have a nice day.

10. Now call the Twilio number to record the voice-mail. Once you have recorded your message your first terminal should be as follows:

```
* Debugger is active!
* Debugger PIN: 269-431-041
127.0.0.1 - - [04/Nov/2020 18:25:04] "GET /answer HTTP/1.1" 200 -
127.0.0.1 - - [04/Nov/2020 18:25:04] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [04/Nov/2020 18:25:25] "POST /answer HTTP/1.1" 200 -
Voicemail recorded successfully...
mp3 File Downloaded...
Uploading File...
File uploaded successfully...
127.0.0.1 - - [04/Nov/2020 18:25:43] "POST /end_call HTTP/1.1" 200 -
```

11. With this done, log into AWS account and navigate to S3 resource. Select your bucket and there should be an mp3 file present. Click on the mp3 file which will open the configurations for that file. In the overview section select the option 'Make Public'. The URL generated is shareable with anyone and no authentication is required. This step is explained by the following figures:



Amazon S3 > recordedvoicemails > voicemail-recording.mp3

voicemail-recording.mp3 Latest version ▼

Overview	Properties	Permissions	Select from
----------	------------	-------------	-------------

Open Download Download as Make public Copy path

Owner
266865473dd4b9521ff8ac96336377c3bd0ed56ef837a835650e7a618db6ce90

Last modified
Nov 4, 2020 10:13:43 PM GMT-0800

Etag
7a330550ef0f4b0d9b592ca5c4ec3148

Storage class
Standard

Server-side encryption
None

Size
36.9 KB

Key
voicemail-recording.mp3

Object URL
<https://recordedvoicemails.s3.us-east-2.amazonaws.com/voicemail-recording.mp3>

12. When this URL is pasted in a browser the system will automatically direct you to download it. Download the file and then you can listen to the voice-mail recording.

Note: This code is made to access the most recent recording, so, if another recording is done then it would overwrite the previous one. This helps in saving space and not creating unnecessary files.

Result

After running the code the following link was generated in the AWS S3 bucket and I have made it public so that it can be accessed by anyone.

<https://recordedvoicemails.s3.us-east-2.amazonaws.com/voicemail-recording.mp3>

References

- [1]. "Communication APIs for SMS, Voice, Video and Authentication," Twilio. [Online].
Available: <https://www.twilio.com/>. [Accessed: 05-Nov-2020].
- [2]. "Amazon Web Services," Amazon. [Online].
Available: <https://aws.amazon.com/>. [Accessed: 05-Nov-2020].
- [3]. "Python documentation," Python. [Online].
Available: <https://www.python.org/doc/>. [Accessed: 05-Nov-2020].