SENDGRID INTEGRATION WITH PYTHON

Date	12 November 2022
Team ID	PNT2022TMID04430
Project Name	Nutrition Assistant Application

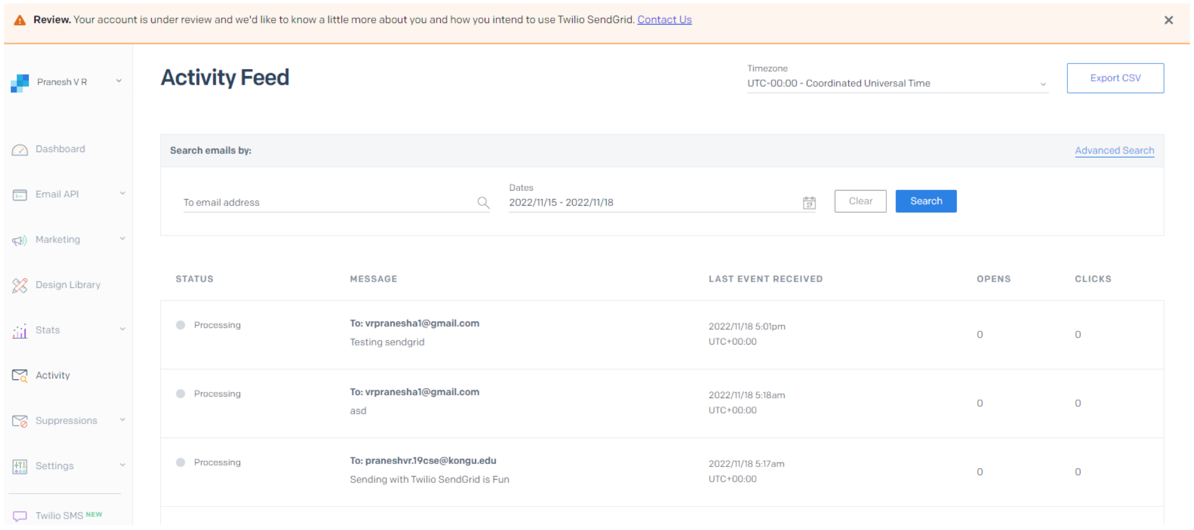
STEP 1:

REQUIREMENTS:

Python 2.6, 2.7, 3.4 or 3.5.

STEP 2:

Create an API key



Review. Your account is under review and we'd like to know a little more about you and how you intend to use Twilio SendGrid. Contact Us

Activity Feed

Timezone: UTC-00:00 - Coordinated Universal Time

Search emails by:

To email address

Dates: 2022/11/15 - 2022/11/18

STATUS	MESSAGE	LAST EVENT RECEIVED	OPENS	CLICKS
Processing	**To: vrpranesha1@gmail.com** Testing sendgrid	2022/11/18 5:01pm UTC+00:00	0	0
Processing	**To: vrpranesha1@gmail.com** asd	2022/11/18 5:18am UTC+00:00	0	0
Processing	**To: praneshvr.19cse@kongu.edu** Sending with Twilio SendGrid is Fun	2022/11/18 5:17am UTC+00:00	0	0

STEP 3:

INSTALL PACKAGE: > pip install sendgrid

SENDGRID PYTHON CODE :

```
def contact():
    if request.method == 'POST':
        from_email = request.form['fromemail']
        to_email = request.form['toemail']
        subject = request.form['subject']
        message = Mail(
            from_email=from_email,
            to_emails=to_email,
            subject=subject,
            html_content='<strong>and easy to do anywhere, even with 
Python</strong>')
        try:
            sg = SendGridAPIClient('SG.GI1duQzZTN-
YP7Fb1TgrGQ.W62v8fADyHvbQFpLBu664cGvjofml_51ViZH_47pG_s')
            response = sg.send(message)
            print(response.status_code)
            print(response.body)
            print(response.headers)
        except Exception as e:
            return render_template('success.html', message = e)
        status = 'Mail sended to ' + to_email+ ' and status is : ' + 
str(response.status_code)
        return render_template('success.html',message = status)
    return render_template('contact.html')
```

HTTP CLIENT PROGRAM

```
from flask import Flask,render_template,request, redirect


import os
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
import requests
import ibm_db, ibm_db_dbi
import pandas as pd
from flask_mail import Mail, Message
from sendgrid import SendGridAPIClient
```

```python
from sendgrid.helpers.mail import Mail

app = Flask(__name__,template_folder="templates")
model=load_model('nutrition.h5')
print("Loaded model from disk")

#ibm DB2 connection
connection = ibm_db.connect('DATABASE=bludb;'
                     'HOSTNAME=8e359033-a1c9-4643-82ef-
8ac06f5107eb.bs2io90l08kqb1od8lcg.databases.appdomain.cloud;'
                     'PORT=30120;'
                     'PROTOCOL=TCPIP;'
                     'SECURITY=SSL;'
                     'SSLServerCertificate=DigiCertGlobalRootCA.crt;'
                     'UID=zsy87446;'
                     'PWD=CoV6JEBhlIYZGUfq;', '', ''
)

print("Connection with ibm db2 is successful.")

# sendgrid_api_key = 'SG.GI1duQzZTN-
YP7Fb1TgrGQ.W62v8fADyHvbQFpLBu664cGvjofml_51ViZH_47pG_s'
# sender_email_address = 'vrpranesha1@gmail.com'
# #Sendgrid requirements
# app.config['SECRET_KEY'] = 'top-secret!'
# app.config['MAIL_SERVER'] = 'smtp.sendgrid.net'
# app.config['MAIL_PORT'] = 587
# app.config['MAIL_USE_TLS'] = True
# app.config['MAIL_USERNAME'] = 'apikey'
# app.config['MAIL_PASSWORD'] = sendgrid_api_key
# app.config['MAIL_DEFAULT_SENDER'] = sender_email_address
# mail = Mail(app)


# Home
@app.route('/')
def home():
    return render_template('index.html')

@app.route('/image1',methods=['GET','POST'])
def image1():
    return render_template("image.html")

#login
```

```python
@app.route('/login', methods=['GET', 'POST'])
def login():
    count = 0
    if request.method == 'POST':
        username = request.form['username']
        password = request.form['password']
        sql = "SELECT USERNAME FROM userdata WHERE USERNAME =
'"+username+"' AND USERPASS = '"+password+"'"
        stmt = ibm_db.exec_immediate(connection, sql)
        while ibm_db.fetch_row(stmt) != False:
          print(ibm_db.result(stmt, 0))
          count = count + 1
        if count == 0:
            return render_template('login.html')
        else:
            return render_template('index.html')


        return render_template('index.html')
        # else:
        #     return render_template("login.html")
    return render_template("login.html")



# Classification and Nutrition Suggestion
@app.route('/predict',methods=['GET', 'POST'])
def launch():
    if request.method=='POST':
        f=request.files['file']
        basepath=os.path.dirname('__file__')
        filepath=os.path.join(basepath,"uploads",f.filename)
        f.save(filepath)

        img=image.load_img(filepath,target_size=(64,64))
        x=image.img_to_array(img)
        x=np.expand_dims(x,axis=0)

        pred=np.argmax(model.predict(x), axis=1)
        print("prediction",pred)
        index=['APPLES','BANANA','ORANGE','PINEAPPLE','WATERMELON']
        result=str(index[pred[0]])

        x=result
        print(x)
        result=nutrition(result)
        # print(result)
        return render_template("0.html",showcase=(result),showcase1=(x))


@app.route('/checkall',methods=['GET','POST'])
def checkall():
    all = ['mango', 'apple', 'pineapple', 'guava', 'grapes', 'dosa',
'idly', 'rice', 'poori', 'tomato', 'carrot']
    data = []
    try:
        for each in all:
            d = nutrition(each)
            data.append(d)
    except:
        pass
    return render_template('checkall.html', data=data)

@app.route('/contact', methods=['GET', 'POST'])
def contact():
    if request.method == 'POST':
        from_email = request.form['fromemail']
        to_email = request.form['toemail']
        subject = request.form['subject']
        message = Mail(
            from_email=from_email,
            to_emails=to_email,
            subject=subject,
            html_content='<strong>and easy to do anywhere, even with
Python</strong>')
        try:
            sg = SendGridAPIClient('SG.GI1duQzZTN-
YP7Fb1TgrGQ.W62v8fADyHvbQFpLBu664cGvjofml_51ViZH_47pG_s')
            response = sg.send(message)
            print(response.status_code)
            print(response.body)
            print(response.headers)
        except Exception as e:
            return render_template('success.html', message = e)
        status = 'Mail sended to ' + to_email+ ' and status is : ' +
str(response.status_code)
        return render_template('success.html',message = status)
    return render_template('contact.html')
```

```
# Nutrition API
def nutrition(index):


    url = "https://calorieninjas.p.rapidapi.com/v1/nutrition"

    querystring = {"query":index}

    headers = {
        'x-rapidapi-key':
"5d797ab107mshe668f26bd044e64p1ffd34jsnf47bfa9a8ee4",
        'x-rapidapi-host': "calorieninjas.p.rapidapi.com"
        }

    response = requests.request("GET", url, headers=headers,
params=querystring)

    # print(response.text)
    return response.json()['items']

@app.route('/success')
def success():
    return render_template('success.html')

if __name__ == "__main__":
   # running the app
    app.run(debug=True)
# using SendGrid's Python Library
# https://github.com/sendgrid/sendgrid-python
```