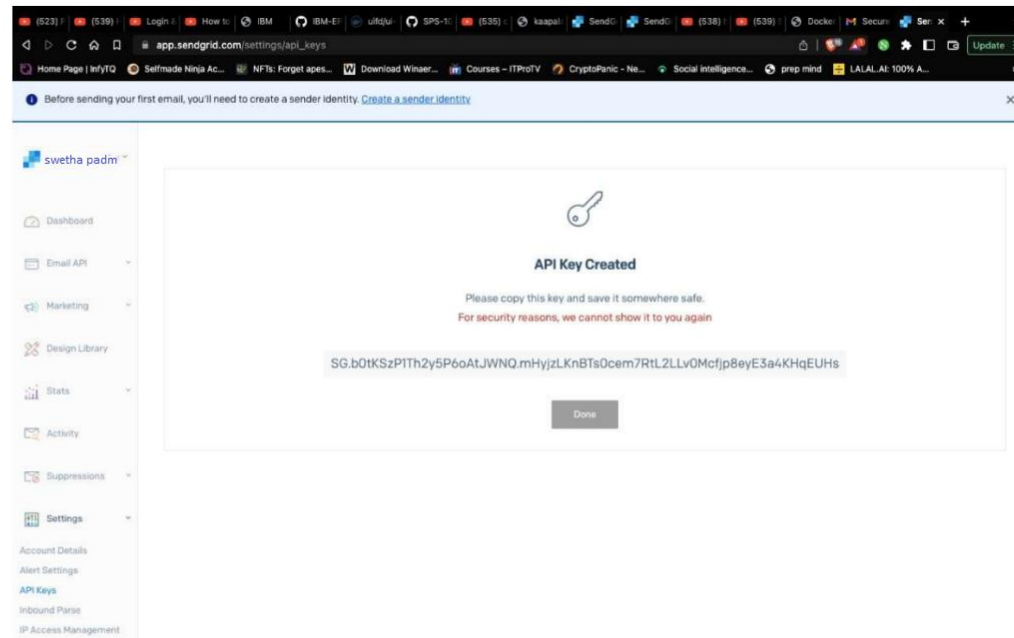


# INTEGRATING SENDGRID SERVICE

## Create An Account Sendgrid Integration With Python Code

Date	14 November 2022
Team ID	PNT2022TMID37166
Project Name	NUTRITION ASSISTANT APPLICATION

Sendgrid Integration With Python Code:



app.sendgrid.com/settings/api\_keys

Home Page | why?Q | Selfmade Ninja Ac... | NFTs: Forget apes... | Download Winrar... | Courses - ITProTV | CryptoPanic - Na... | Social Intelligence... | prep mind | LALALAL: 100% A...

Before sending your first email, you'll need to create a sender identity. [Create a sender identity](#)

swetha padman

API Keys

Create API Key

NAME	API KEY	ACTION
swetha	API Key ID: b0KISzPTth2y6P6oALJWNQ	<div><div></div><div>*****</div><div></div></div>

The image shows a Visual Studio Code editor window with a dark theme. On the left, the Explorer sidebar displays a file tree for a project named 'UNTITLED (WORKSPACE)'. The tree includes a 'flask' folder with subfolders like '\_\_pycache\_\_', 'env', 'static', and 'templates'. The 'templates' folder contains 'aduthavan.html', 'home.html', and 'index.html'. Other files in the project include 'admin.data.csv', 'admin.csv', 'aduthavan.py', 'app.py', 'Cert.crt', 'cookie.py', 'db2.py', 'dockerfile', 'dockerfile.txt', 'next.py', 'requirements.txt', 'tempCodeRunnerFile.python', 'test.py', 'admin page', and 'kaapalan\_001'. The 'app.py' file is selected and open in the main editor. The code in 'app.py' is a Flask application that uses SendGrid for email sending. It imports necessary modules like 'sendgrid.helpers.mail', 'flask\_mysqldb', 'logging', 'jinja2', 'bcrypt', and 'json'. It initializes a Flask app with a secret key 'a'. A database connection is established using 'ibm\_db'. A 'send\_mail' function is defined to send emails via SendGrid. The function constructs a message with a 'from\_email' of 'venkatkissshore2005@gmail.com', a 'to\_email' of 'email', a subject of 'caution', a plain text content of 'Please Stay Safe', and an HTML content of '<h2>You are entering into a containment Zone</h2>'. The function then uses the 'SendGridAPIClient' to send the message and prints the response details. An exception handler is present for any errors. The main route '/' is decorated with '@app.route' and calls the 'home' function, which renders the 'home.html' template. The status bar at the bottom indicates the current cursor position is at line 82, column 27, with 1086 characters selected. It also shows the file encoding as UTF-8, line endings as LF, and the Python interpreter as 3.10.8 ('env': venv).

```
flask > app.py > ...
5 from sendgrid.helpers.mail import Mail
6 from flask_mysqldb import MySQL
7 from logging import error
8 from jinja2.utils import select_autoescape
9 import bcrypt
10 import json
11
12
13 app = Flask(__name__)
14
15 app.secret_key = 'a'
16
17 conn = ibm_db.connect("DATABASE=bludb;HOSTNAME=ba99a9e6-d59e-4883-8fc0-d6a8c9f7a08f.clogj3sd0tgtu0lq
18 def send_mail(email):
19     print(email)
20     message = Mail(from_email='venkatkissshore2005@gmail.com',
21                    to_emails=email,
22                    subject='caution',
23                    plain_text_content='Please Stay Safe',
24                    html_content='<h2>You are entering into a containment Zone</h2>')
25
26     try:
27         sg = SendGridAPIClient(
28             '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')
29         response = sg.send(message)
30         print(response.status_code)
31         print(response.body)
32         print(response.headers)
33     except Exception as e:
34         print(e)
35 @app.route('/')
36
37 def home():
38     return render_template('home.html')
39
40
41
42
```