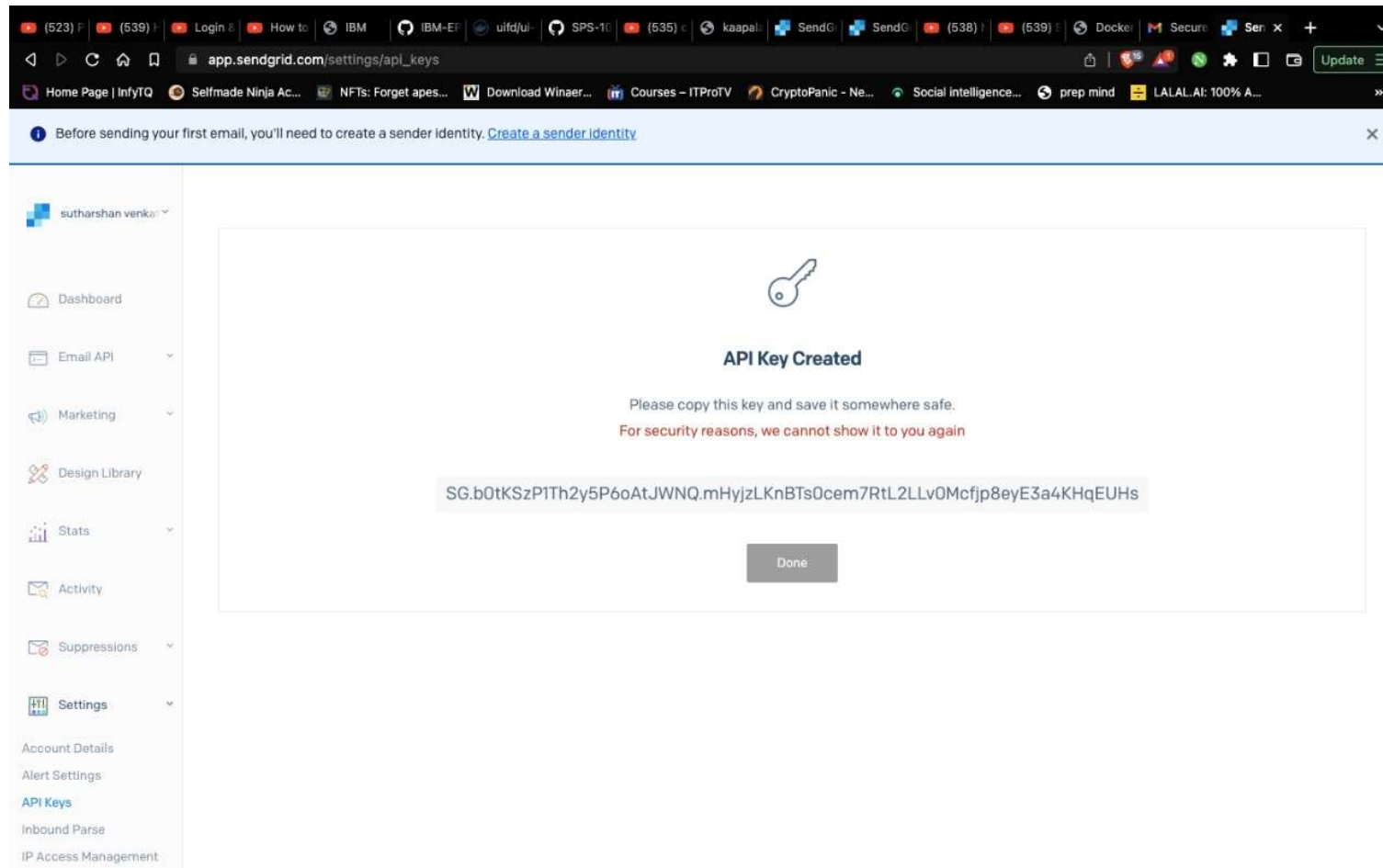


INTEGRATING SENDGRID SERVICE

Create An Account Sendgrid Integration With Python Code

Date	14 November 2022
Team ID	PNT2022TMID37165
Project Name	CONTAINMENT ZONE ALERTING

Sendgrid Integration With Python Code:



The screenshot shows a web browser window with the URL `app.sendgrid.com/settings/apl_keys`. The browser's address bar and tabs are visible at the top. A blue notification banner at the top of the page reads: "Before sending your first email, you'll need to create a sender identity. [Create a sender identity](#)".

The main content area displays a success message for creating an API key. At the top is a key icon. Below it, the text "API Key Created" is centered. A message follows: "Please copy this key and save it somewhere safe. For security reasons, we cannot show it to you again". The API key itself is displayed in a light blue box: `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`. A "Done" button is located below the key.

The left sidebar contains a navigation menu with the following items: "sutharshan venka...", "Dashboard", "Email API", "Marketing", "Design Library", "Stats", "Activity", "Suppressions", "Settings", "Account Details", "Alert Settings", "API Keys" (highlighted), "Inbound Parse", and "IP Access Management".

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

sutharshan venka

Dashboard

Email API

Marketing

Design Library

Stats

Activity

Suppressions

Settings

Account Details

Alert Settings

API Keys

Inbound Parse

IP Access Management

API Keys

Create API Key

NAME	API KEY	ACTION
sutharshan API Key ID: b0tKSzP1Th2y5P6oAtJWNQ	*****	

The screenshot shows a VS Code editor with a dark theme. On the left is the Explorer sidebar showing a file tree for a project named 'UNTITLED (WORKSPACE)'. The tree includes a 'flask' folder with subfolders like 'static' and 'templates'. The 'templates' folder contains 'aduthavan.html', 'home.html', and 'index.html'. Other files include 'admin data.csv', 'admin.csv', 'aduthavan.py', 'app.py' (selected), 'Cert.crt', 'cookie.py', 'db2.py', 'dockerfile', 'dockerfile.txt', 'next.py', 'requirements.txt', 'tempCodeRunnerFile.python', 'test.py', 'admin page', and 'kaapalan_001'. The main editor area shows the code for 'app.py'. The code imports Flask, SendGrid, Flask-MySQL, logging, Jinja2, bcrypt, and json. It initializes a Flask app, sets a secret key, and connects to an IBM DB. A 'send_mail' function is defined to send emails via SendGrid. A route '/' is defined to render the 'home.html' template. The status bar at the bottom indicates 'Ln 82, Col 27 (1086 selected)', 'Spaces: 4', 'UTF-8', 'LF', 'Python 3.10.8 (env: venv)', and 'Go Live'.

```
flask > app.py > ...
5 from sendgrid.helpers.mail import Mail
6 from flask_mysql 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='venkatkisshore2005@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
```