



PI DATA STRATEGY & CONSULTING

Data Engineering

Send email notification using Sendgrid and Databricks

Autor: Ingaramo, Iván

Guía

1- Crear cuenta en Twilio Sendgrid

2- Generar autenticación de 2 pasos

3- Crear una API Key

4- Crear un Single Sender

5- Instalar la librería Sendgrid a nivel de Cluster en Databricks usando PyPi

6- Código para utilizar la API de Sendgrid (Python)



PI DATA STRATEGY & CONSULTING

1- Crear cuenta en Twilio Sendgrid

- Crear cuenta: <https://signup.sendgrid.com>
- Python Quickstart: <https://docs.sendgrid.com/for-developers/sending-email/quickstart-python>
- Consideración:
 - 100 emails por día gratis



2- Generar autenticación de 2 pasos

- Descargar la app Authy (mobile):
<https://authy.com/download/>
- En el Dashboard principal de la página:
 - Settings -> Two Factor Authentication -> Authy App
- En la app:
 - Agregar cuenta
 - Escanear Código QR



3- Crear una API Key

- En el Dashboard principal de la página:
 - Settings -> API Key-> Create API Key
 - Seleccionar permisos
 - Guardar la Key en un lugar seguro, sólo se muestra una vez. (Variable de Entorno, Key Vault, etc)



4- Crear un Single Sender

- En el Dashboard principal de la página:
 - Settings -> Sender Authentication -> Verify a Single Sender
 - Seleccionar “Create New Sender”



5- Instalar la librería Sendgrid a nivel de Cluster en Databricks usando PyPi

- Librería Sendgrid: <https://pypi.org/project/sendgrid/>
- En Databricks:
 - Compute -> Cluster -> Libraries -> Install New
- Seleccionar PyPi como Library Source



6- Código para utilizar la API de Sendgrid (Python)

```
import sendgrid

import os

from sendgrid.helpers.mail import Mail, Email, To, Content


sg = sendgrid.SendGridAPIClient(api_key=os.environ.get('SENDGRID_API_KEY'))

from_email = Email("test@example.com") # Change to your verified sender
to_email = To("test@example.com") # Change to your recipient
subject = "Sending with SendGrid is Fun"
content = Content("text/plain", "and easy to do anywhere, even with Python")
mail = Mail(from_email, to_email, subject, content)


# Get a JSON-ready representation of the Mail object

mail_json = mail.get()


# Send an HTTP POST request to /mail/send

response = sg.client.mail.send.post(request_body=mail_json)

print(response.status_code)

print(response.headers)
```



PI DATA STRATEGY & CONSULTING