# **Access mail remotely google API**

In this project, we have established a connection between our Gmail account and a Google API, enabling us to interact with the attachment (CSV file) present in the initial email.

1. Write a code in python to remotely access the data in first inbox of your mail and store the attachment in it , in the form of a dataframe and manipulate the dataframe. (training.py)
2. Write a code in python where you identify the hyperlink in the email and save that file as a dataframe on local desktop. (main.py)

## **[This code is written on MacOS and there will be changes for Windows ]**

## **Step 1: Set Up the Gmail API and Obtain Credentials**

1. Go to the Google Cloud Console (<https://console.cloud.google.com/>).
2. Create a new project (if you don't have one) or select an existing project.
3. In the left navigation menu, go to "APIs & Services" > "Library."
4. Search for "Gmail API" and enable it for your project.
5. In the left navigation menu, go to "APIs & Services" > "Credentials."
6. Click on "Create credentials" and select "OAuth client ID."
7. Choose "Desktop app" as the application type and give it a name.
8. Click "Create" to generate the OAuth client ID and client secret.
9. Download the client secret JSON file and save it to your computer.

## **Step 2: Install Required Packages**

Ensure you have the required Python packages installed. Open your terminal or command prompt and run the following command to install them if they are not already installed:

### **Copy code**

pip install pandas google-auth google-auth-oauthlib

## **Step 3: Replace Client Secret Path**

Replace 'path\_to\_your\_client\_secret.json' in the code with the actual path to the client secret JSON file you downloaded in Step 1.

## **Step 4: Authenticate with Gmail API**

Run the code, and it will prompt you to visit a URL in your web browser. Open the URL, and you'll be asked to log in to your Google account and authorize the application.

## **Step 5: Run the Code**

Once you have completed the authentication process, the code will execute. It will fetch the first email from your inbox, check if there is a CSV attachment, load the CSV data into a DataFrame, print the column names and the original DataFrame, drop the first and second rows, and save the modified DataFrame as a new CSV file named 'modified\_dataframe.csv' on your desktop.

## **Step 6: Verify the Output**

After the code execution, check the output in your terminal or command prompt to see the printed column names, original DataFrame, and modified DataFrame. Additionally, you should find the 'modified\_dataframe.csv' file on your desktop containing the modified DataFrame in CSV format.

