Gmail API Setup Walkthrough (Mac & Windows Users)

1. Project Setup in Google Cloud Console:

* Go to the Google Cloud Console: [Google Cloud Console](<https://console.cloud.google.com/>).
* Create a new project or select an existing one.
* Enable the Gmail API: Search for "Gmail API" in the API Library and enable it for your project.

1. Create Credentials:

* In the Google Cloud Console, navigate to "APIs & Services" -> "Credentials."
* Click "Create Credentials" and select "OAuth client ID."
* Configure the OAuth consent screen (if you haven't already). This includes providing an application name, user support email, and contact information.
* Choose the application type:
* For Windows & Mac users running a local Python script, select "Desktop app."
* Give your OAuth client a name (e.g., "Gmail API Client").
* Create the credentials. This will generate a Client ID and Client Secret.

1. Download Credentials JSON:

* After creating the OAuth client ID, you should have the option to download the credentials as a JSON file.
* This file is usually named credentials.json and contains your Client ID and Client Secret, which are essential for authenticating with the Gmail API.
* Store credentials.json securely and never share it publicly!

1. Install the Google API Client Library for Python:

Open a terminal (Mac) or command prompt (Windows) and run:

pip install --upgrade google-api-python-client google-auth-httplib2 google-auth-oauthlib

1. Authentication Flow in Your Python Script:

When you run your script for the first time, it will:

* Read the credentials.json file.
* Use the google-auth-oauthlib library to start an OAuth 2.0 flow.
* Open a browser window and ask you to grant your application permission to access your Gmail account. This will be the Gmail account you want to check for phishing.
* After permission is granted, the script will receive an access token and a refresh token.
* Save these tokens in a file named token.json to avoid requiring authorization every time you run the script.