

Google API Documentation

Download Json file
Change Database URL ID

Link: <https://developers.google.com/docs/api/quickstart/python>

- 1) Follow above link to enable Google API and create JSON credential
- 2) Enable the API

Enable the API

Before using Google APIs, you need to turn them on in a Google Cloud project. You can turn on one or more APIs in a single Google Cloud project.

- In the Google Cloud console, enable the Google Docs API.

[Enable the API](#)

3) Perform the following to create the credential.json

Authorize credentials for a desktop application

To authenticate as an end user and access user data in your app, you need to create one or more OAuth 2.0 Client IDs. A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, you must create a separate client ID for each platform.

1. In the Google Cloud console, go to Menu ☰ > **APIs & Services** > **Credentials**.

[Go to Credentials](#)

2. Click **Create Credentials** > **OAuth client ID**.
3. Click **Application type** > **Desktop app**.
4. In the **Name** field, type a name for the credential. This name is only shown in the Google Cloud console.
5. Click **Create**. The OAuth client created screen appears, showing your new Client ID and Client secret.
6. Click **OK**. The newly created credential appears under **OAuth 2.0 Client IDs**.
7. Save the downloaded JSON file as `credentials.json`, and move the file to your working directory.

3) Perform the following to create the credential.json

Authorize credentials for a desktop application

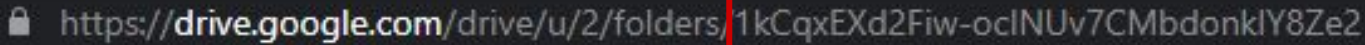
To authenticate as an end user and access user data in your app, you need to create one or more OAuth 2.0 Client IDs. A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, you must create a separate client ID for each platform.

1. In the Google Cloud console, go to Menu ☰ > **APIs & Services** > **Credentials**.

[Go to Credentials](#)

2. Click **Create Credentials** > **OAuth client ID**.
3. Click **Application type** > **Desktop app**.
4. In the **Name** field, type a name for the credential. This name is only shown in the Google Cloud console.
5. Click **Create**. The OAuth client created screen appears, showing your new Client ID and Client secret.
6. Click **OK**. The newly created credential appears under **OAuth 2.0 Client IDs**.
7. Save the downloaded JSON file as `credentials.json`, and move the file to your working directory.

- 4) Make sure the JSON file name matches
- 5) Change **CLIENT_SECRET_FILE** to the absolute path of JSON file
- 6) Change the **folder_database_id** variable to select the folder to upload data (copy the last string after '/' on the URL) as shown below)

A screenshot of a Google Drive URL: `https://drive.google.com/drive/u/2/folders/1kCqxEXd2Fiw-ocINUv7CMbdonkly8Ze2`. The folder ID `1kCqxEXd2Fiw-ocINUv7CMbdonkly8Ze2` is highlighted with a red rectangular box.

`compute_export_enf.py`

```
13  # Constants for file locations
14  folder_archive = 'Archived_Recordings/'
15  folder_audio = 'Power_Recordings/'
16  folder_enf = 'ENF_Data/'
17
18  # Create google drive service instance
19  CLIENT_SECRET_FILE = '/home/pi/Desktop/pyenf_extraction/Power_Rec_Feature/credentials.json'
20  API_NAME = 'drive'
21  API_VERSION = 'v3'
22  SCOPES = ['https://www.googleapis.com/auth/drive']
23  folder_database_id = '1kCqxEXd2Fiw-ocINUv7CMbdonkly8Ze2' #upload to database folder
24  dev = 'Dev3'
25
26  # Create Google Drive service instance
27  service = Create_Service(CLIENT_SECRET_FILE, API_NAME, API_VERSION, SCOPES)
```

Change dev to
specify the
prefix for the
filename

Raspberry Pi

1. If the Pi is automated, the terminal always execute the autorun.sh script on the Desktop
 - a. Comment out the last line in .bashrc (sudo nano .bashrc)
 - b. Source autorun.sh executes the script
2. After updating the compute_export_enf.py code, make sure to run:
 - a. Source autorun.sh
 - i. Make sure the compute_export_enf.py code can access the Google drive
3. When new credential json is changed, the program will request you to manual log into your gmail account to authenticate
 - a. Only needs to be done ONCE