First, install the required libraries:

pip install gspread google-auth google-auth-oauthlib google-auth-httplib2 pandas

Then, create a **credentials.json** file with your Google Sheets API credentials. You can follow this guide on how to obtain the credentials: <https://developers.google.com/sheets/api/quickstart/python>

1. Open your Google Sheet in your web browser.
2. Take a look at the URL in the address bar. It should look something like this:

<https://docs.google.com/spreadsheets/d/your_sheet_key_here/edit#gid=0>

1. The sheet key is the long string of letters and numbers between "/d/" and "/edit" in the URL. In the example above, it's **your\_sheet\_key\_here**.

Here is a step-by-step guide to create a **credentials.json** file for accessing the Google Sheets API:

1. Go to the [Google Cloud Console](https://console.cloud.google.com/).
2. If you haven't already, sign in with your Google account.
3. Click on the project drop-down (top right corner, next to the "Google Cloud Platform" logo) and select or create the project you want to use for accessing the Google Sheets API.
4. Click the hamburger menu icon (three horizontal lines) in the top left corner of the page, and then click "APIs & Services" > "Dashboard".
5. Click "+ ENABLE APIS AND SERVICES" at the top of the page.
6. In the API Library, search for "Google Sheets API" using the search bar.
7. Click on "Google Sheets API" in the search results.
8. Click the "ENABLE" button to enable the Google Sheets API for your project.
9. After the API is enabled, click "Create credentials" in the top-right corner of the page.
10. In the "Add credentials to your project" form, follow these steps:
    * Which API are you using? - Select "Google Sheets API".
    * Where will you be calling the API from? - Choose "Other non-UI (e.g. cron job, daemon)".
    * What data will you be accessing? - Select "Application data".
    * Click the "What credentials do I need?" button.
11. In the "Create a service account" form:
    * Enter a name for your service account (e.g., "sheet-reader").
    * Add a description (optional) and click "Create".
12. On the "Grant this service account access to project" step, set the role to "Editor" to allow read and write access to the Google Sheets. If you only need read access, you can choose the "Viewer" role instead. Then click "Continue".
13. On the "Grant users access to this service account" step, you can skip this and click "Done".
14. You will be redirected to the "Service accounts" page. Find the service account you just created and click the Edit icon (a pencil) on the right side.
15. In the "Service account details" page, click "Add Key" and select "JSON".
16. A JSON file will be generated and downloaded to your computer. This is your **credentials.json** file.

Now you have your **credentials.json** file ready to use in your Python script. Make sure to place the file in the same directory as your Python script, or update the file path in the script accordingly.

Top of Form

Bottom of Form

Follow these steps to grant the necessary permissions:

1. Open the **credentials.json** file in a text editor.
2. Locate the **client\_email** field, which should look something like this:

"client\_email": "your-service-account-email@your-project-id.iam.gserviceaccount.com",

1. Copy the email address associated with the **client\_email** field.
2. Open the Google Sheet you want to access with the script.
3. Click the "Share" button in the top right corner of the Google Sheets interface.
4. In the "Share with people and groups" input field, paste the service account email you copied earlier.
5. Make sure the service account has either "Viewer" or "Editor" permissions, depending on your needs.
6. Click "Done" to save the changes.

After granting the service account access to your Google Sheet, try running the script.

Next, you can use the following script to read the data from the Google Sheet and create a cleaner dataset:

import pandas as pd

import gspread

from google.oauth2 import service\_account

# Load credentials and connect to Google Sheets API

scope = ['https://www.googleapis.com/auth/spreadsheets']

creds = service\_account.Credentials.from\_service\_account\_file('credentials.json', scope)

client = gspread.authorize(creds)

# Replace this with your Google Sheet key

sheet\_key = '1fS9kNDh1DlJiyphvD\_ZbpSERU282xsUPJmjx4iUtm4I'

sheet\_name = 'Sheet1'

# Read the data from the Google Sheet

sheet = client.open\_by\_key(sheet\_key).worksheet(sheet\_name)

data = sheet.get\_all\_records()

# Convert the data to a pandas DataFrame

df = pd.DataFrame(data)

# Clean the dataset

# You can perform any cleaning operation here based on your requirements

clean\_df = df.dropna() # Example: remove rows with missing values

# Save the cleaned dataset to a new CSV file

clean\_df.to\_csv('clean\_dataset.csv', index=False)

print("Cleaned dataset saved to clean\_dataset.csv")