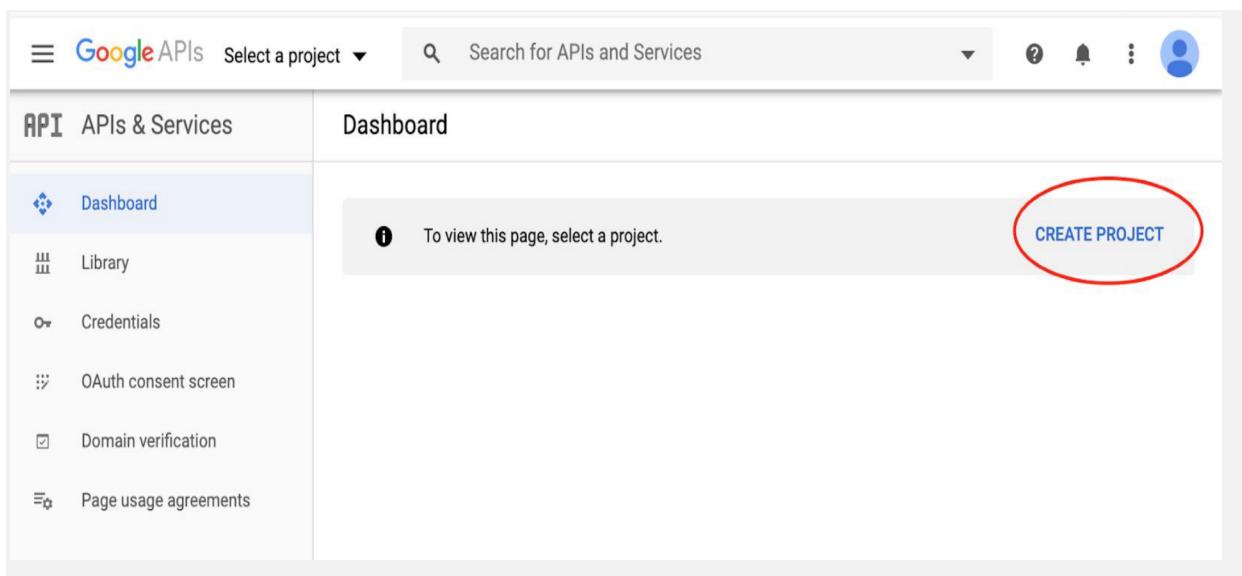


## Get Authentication for Google Service API

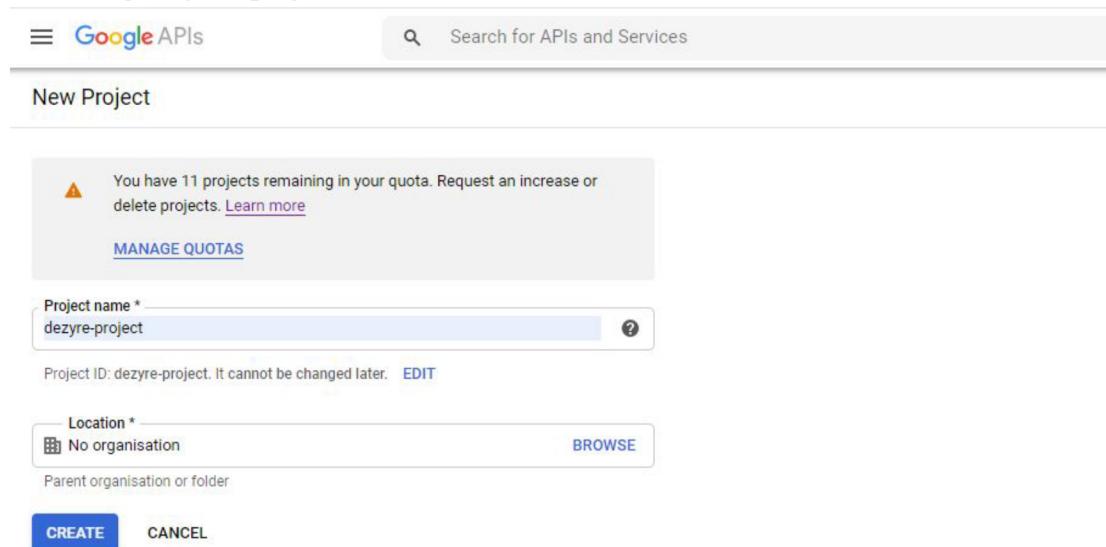
First, we need to get the authentication files for Google Service API, so our Python code can access Google Drive. To do that, we need to:

1. Create a new project in [Google Developer Console](#) by clicking “CREATE PROJECT” as follows.



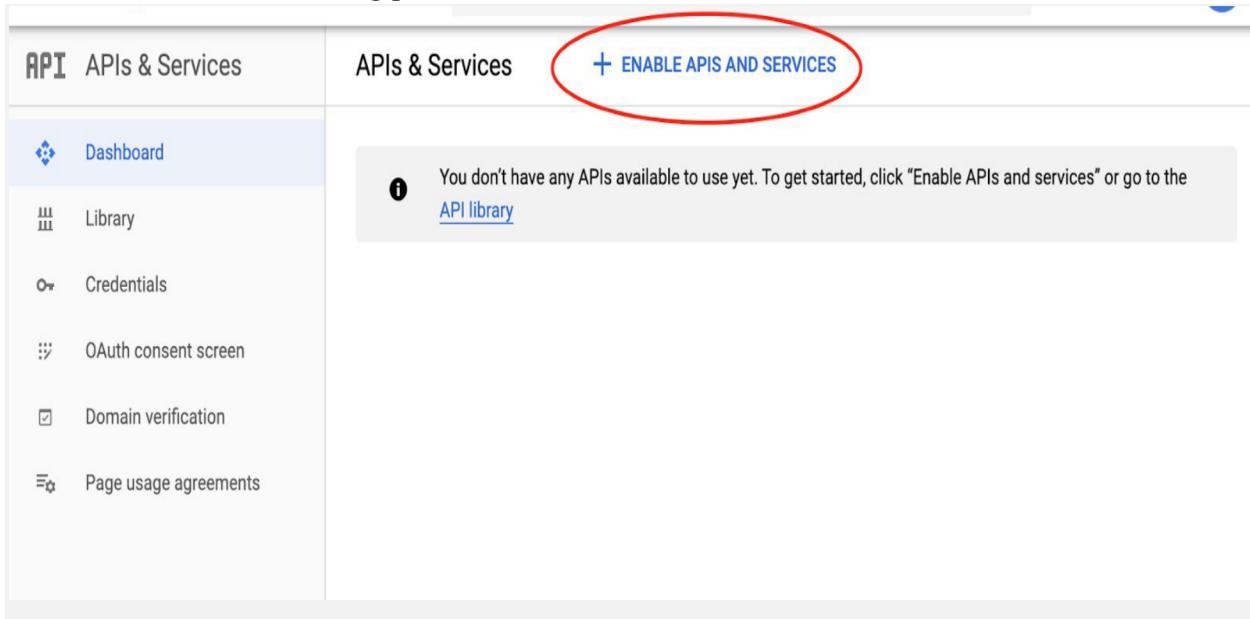
The screenshot shows the Google APIs Dashboard. On the left, there is a sidebar with the following options: Dashboard (selected), Library, Credentials, OAuth consent screen, Domain verification, and Page usage agreements. The main content area has a message: “To view this page, select a project.” To the right of this message is a blue button labeled “CREATE PROJECT”, which is circled in red. The top navigation bar includes the Google APIs logo, a “Select a project” dropdown, a search bar, and user account icons.

You can give your project a name or leave it as default.



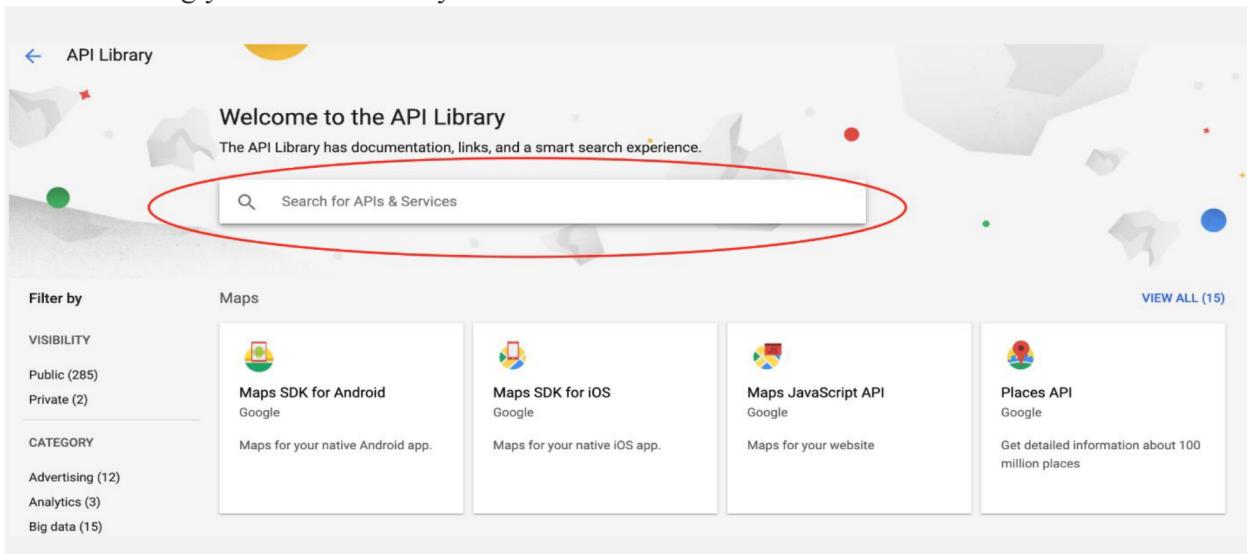
The screenshot shows the “New Project” creation page. At the top, there is a message: “You have 11 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)”. Below this is a “MANAGE QUOTAS” link. The main form fields are: “Project name \*” with the value “dezyre-project”, “Location \*” with the value “No organisation”, and a “BROWSE” button. Below the location field, there is a note: “Parent organisation or folder”. At the bottom of the form are two buttons: a blue “CREATE” button and a “CANCEL” button.

2. Enable APIs and Services by clicking the “ENABLE APIs AND SERVICES” as indicated by the red circle in the following picture.



The screenshot shows the Google Cloud Platform APIs & Services page. On the left is a sidebar with links: Dashboard, Library, Credentials, OAuth consent screen, Domain verification, and Page usage agreements. The main content area is titled 'APIs & Services' and contains a message: 'You don't have any APIs available to use yet. To get started, click "Enable APIs and services" or go to the API library'. A red circle highlights the '+ ENABLE APIs AND SERVICES' button.

That will bring you the API library as below.



The screenshot shows the Google Cloud Platform API Library page. At the top, there is a search bar with the placeholder 'Search for APIs & Services'. A red circle highlights this search bar. Below the search bar, there are filters for 'VISIBILITY' (Public 285, Private 2) and 'CATEGORY' (Advertising 12, Analytics 3, Big data 15). The main content area is titled 'Maps' and shows four API cards: 'Maps SDK for Android' (Google), 'Maps SDK for iOS' (Google), 'Maps JavaScript API' (Google), and 'Places API' (Google). Each card has a small icon and a brief description. A 'VIEW ALL (15)' link is located in the top right corner of the list.

Search “Google Drive” in the API library (indicated by the red circle in the picture). You’ll get the following snapshot.

← Search Google Drive X

Filter by

CATEGORY

- Analytics (1)
- Big data (2)
- Developer tools (1)
- G Suite (1)
- Healthcare (1)
- Mobile (1)
- Storage (1)

5 results

 Google Drive API  
Google  
The Google Drive API allows clients to access resources from Google Drive

 Drive Activity API  
Google  
Provides a historical view of activity in Google Drive.

 Cloud Healthcare API  
Google  
Store and access healthcare data on Google Cloud Platform.

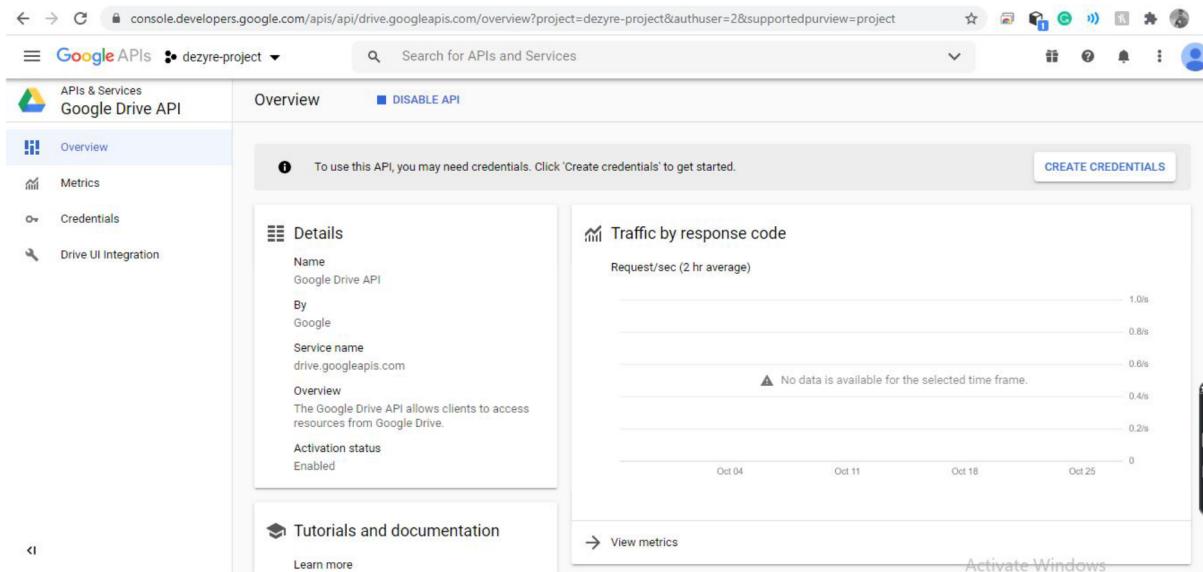
Click the “Google Drive API” icon and it will bring you to the next step as follows.

← API Library

 Google Drive API  
Google  
The Google Drive API allows clients to access resources from Google Drive

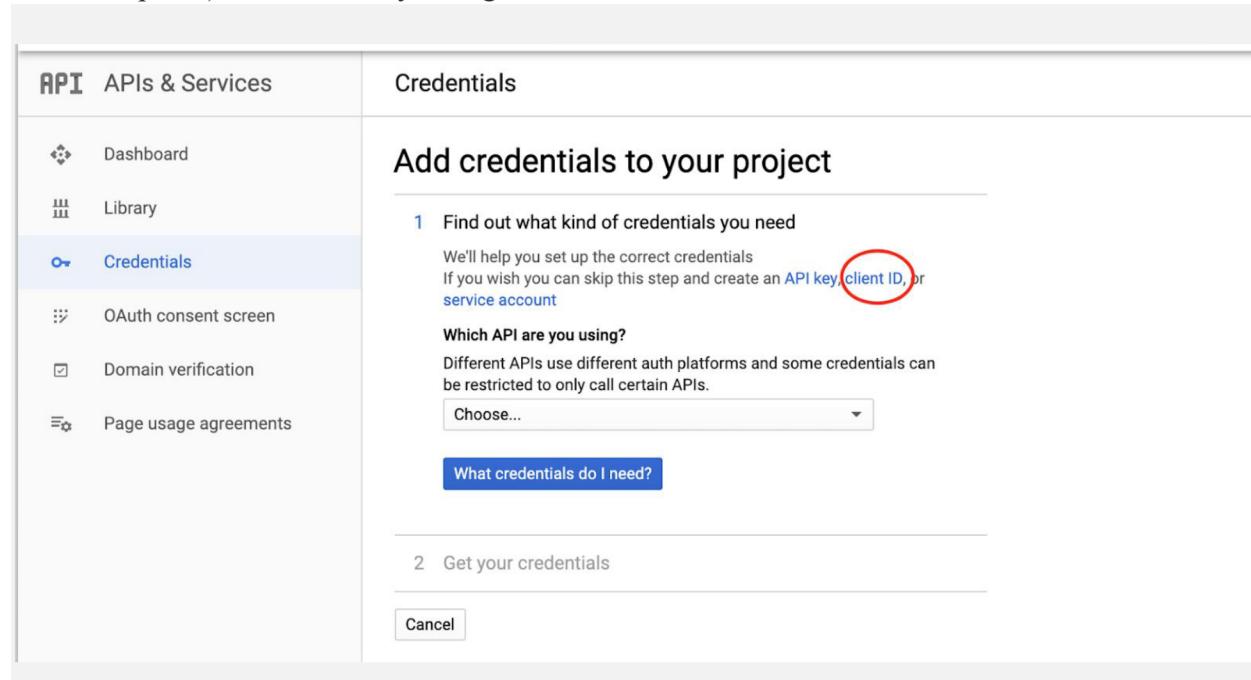
ENABLE TRY THIS API ▾

Then click “ENABLE”, which will enable your Google Drive API service. You’ll get to the next step as follows.



The screenshot shows the Google Developers API console for the project 'dezyre-project'. The 'Google APIs' section is selected. On the left, the 'Overview' tab is active, showing the 'Google Drive API' service. The 'Details' section shows the service is 'Enabled'. To the right, there's a 'Traffic by response code' chart and a 'Tutorials and documentation' section. A red circle highlights the 'CREATE CREDENTIALS' button in the top right corner of the main content area.

3. Create credentials by clicking the “CREATE CREDENTIALS” icon (indicated by red circle in above snapshot). Here’s what you’ll get.



The screenshot shows the 'Add credentials to your project' page. The left sidebar is titled 'API APIs & Services' and includes 'Dashboard', 'Library', 'Credentials' (which is selected and highlighted in blue), 'OAuth consent screen', 'Domain verification', and 'Page usage agreements'. The main content area is titled 'Add credentials to your project' and contains two steps: 1. 'Find out what kind of credentials you need' (with a note about API keys, client ID, or service account) and 2. 'Get your credentials'. A red circle highlights the 'client ID' link in the first step's text.

In the above snapshot, we need to click “client ID” as that’s the Python program needs. Then click “CREATE” and download the JSON file as shown by the following snapshots.

**API** APIs & Services

- [Dashboard](#)
- [Library](#)
- Credentials**
- [OAuth consent screen](#)
- [Domain verification](#)
- [Page usage agreements](#)

[Create OAuth client ID](#)

A client ID is used to identify a single app to Google's OAuth servers. If your app runs on multiple platforms, each will need its own client ID. See [Setting up OAuth 2.0](#) for more information.

**Application type \***

[Learn more about OAuth client types](#)

**Name \***

The name of your OAuth 2.0 client. This name is only used to identify the client in the console and will not be shown to end users.

CREATE
CANCEL

<b>API</b> APIs & Services		Credentials									
		<a href="#">+ CREATE CREDENTIALS</a>		<a href="#">DELETE</a>							
<small>Create credentials to access your Google services</small>											
<b>API Keys</b>											
		<input type="checkbox"/>	Name	Creation date	Restrictions	Key					
			Desktop client 1	May 30, 2020	Desktop	874591480368-ik8p...					
No API keys to display											
<b>OAuth 2.0 Client IDs</b>											
		<input type="checkbox"/>	Name	Creation date	Type	Client ID					
			Desktop client 1	May 30, 2020	Desktop	874591480368-ik8p...					
<a href="#">Download</a>											
<b>Service Accounts</b>											
		<input type="checkbox"/>	Email	Name	<a href="#">Usage with all services (last 30 days)</a>						

The downloaded JSON file is the one we need for our Python code to access Google Drive.

### Install PyDrive Module :

Once you have the JSON file to access Google Drive, we can install a Python library — [PyDrive](#) using `pip install pydrive`.

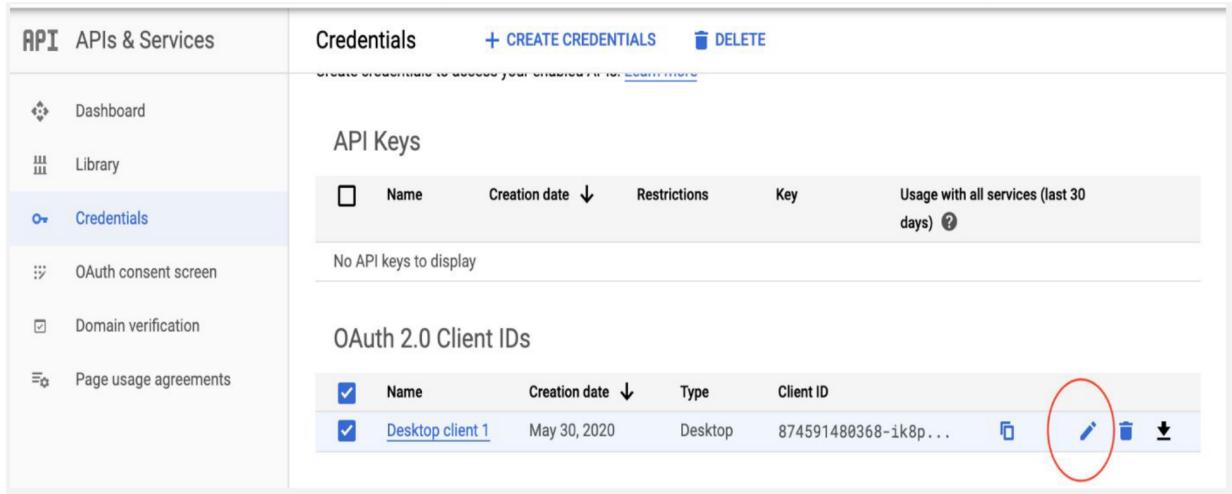
The google authentication code will do authentication and list all files in your Google Drive. Note that every time you run the following program, the code will open a web browser to ask you to input your Google account and password.

To avoid inputting passwords every time, we can create a `settings.yaml` file to save all the credentials. The details can be found from the [PyDrive official document](#). The yaml file is like the following.

`settings.yaml` file :

```
client_config_backend: settings
client_config:
  client_id: your_client_id
  client_secret: your_client_secret
  save_credentials: True
  save_credentials_backend: file
  save_credentials_file: credentials.json
  get_refresh_token: True
  oauth_scope:
    - https://www.googleapis.com/auth/drive.file
```

The `client_id` and `client_secret` can be found by clicking the editing icon in the following picture.



The screenshot shows the Google Cloud Platform API & Services Credentials page. The left sidebar has 'APIs & Services' selected, with 'Dashboard', 'Library', 'Credentials' (which is highlighted in blue), 'OAuth consent screen', 'Domain verification', and 'Page usage agreements'. The main area is titled 'Credentials' with buttons for '+ CREATE CREDENTIALS' and 'DELETE'. A sub-section titled 'API Keys' shows a table with one row: 'No API keys to display'. Below it is a section titled 'OAuth 2.0 Client IDs' with a table showing one entry:

<input checked="" type="checkbox"/>	Name	Creation date	Restrictions	Key	Usage with all services (last 30 days)
<input checked="" type="checkbox"/>	<a href="#">Desktop client 1</a>	May 30, 2020	Desktop	874591480368-ik8p...	<a href="#"><span style="border: 1px solid red; border-radius: 50%; padding: 2px;">Edit</span></a> <a href="#"><span style="border: 1px solid red; border-radius: 50%; padding: 2px;">Delete</span></a> <a href="#"><span style="border: 1px solid red; border-radius: 50%; padding: 2px;">Download</span></a>

Rerun the above Python code, the program will ask you to input your Google password again. Then it will create a `credentials.json` file. Next time, Python will just pick up that file to finish authentication automatically. Therefore, you don't need to type your password again.