Using Python to Extract Data from YouTube

What We Will Need

- Google API’s Python client package and unidecode package
- Authentication information for Google APIs

Install Package

- We will install and use a python package provided for Google API-Related product/service development.

    $ pip install --upgrade google-api-python-client

    reference: https://developers.google.com/api-client-library/python/

- Another package we install is unidecode, a package that handles unicode-ascii translation.

    $ pip install unidecode

    reference: https://pypi.python.org/pypi/Unidecode

Google APIs Set-up

Google APIs (https://developers.google.com/api-client-library/python/start/get_started#setup)

If you have never created a Google APIs Console project, read the Managing Projects page and create a project in the Google Developers Console.

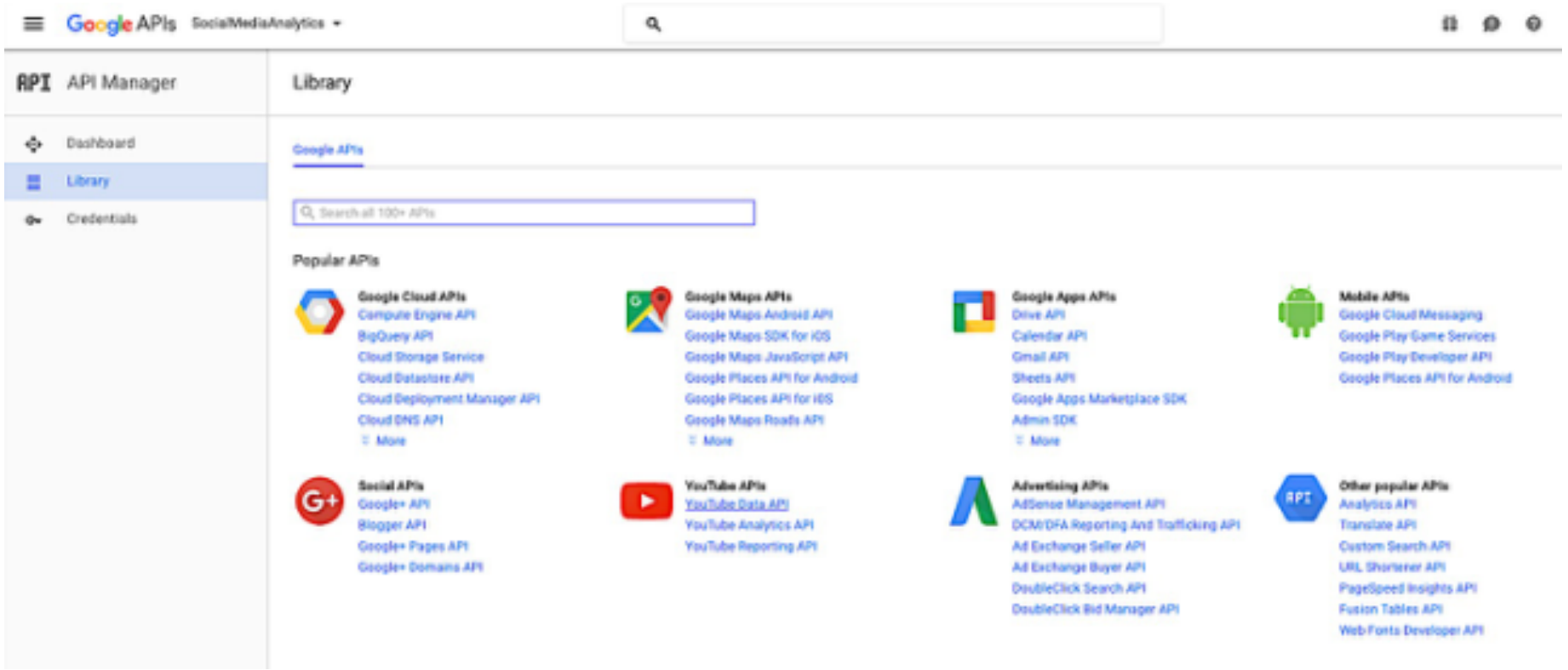
https://console.developers.google.com/apis/library

For example, I created a project named *SocialMedaiAnalytics* and enabled YouTube Data API v3.

See the instructions in the next slides.

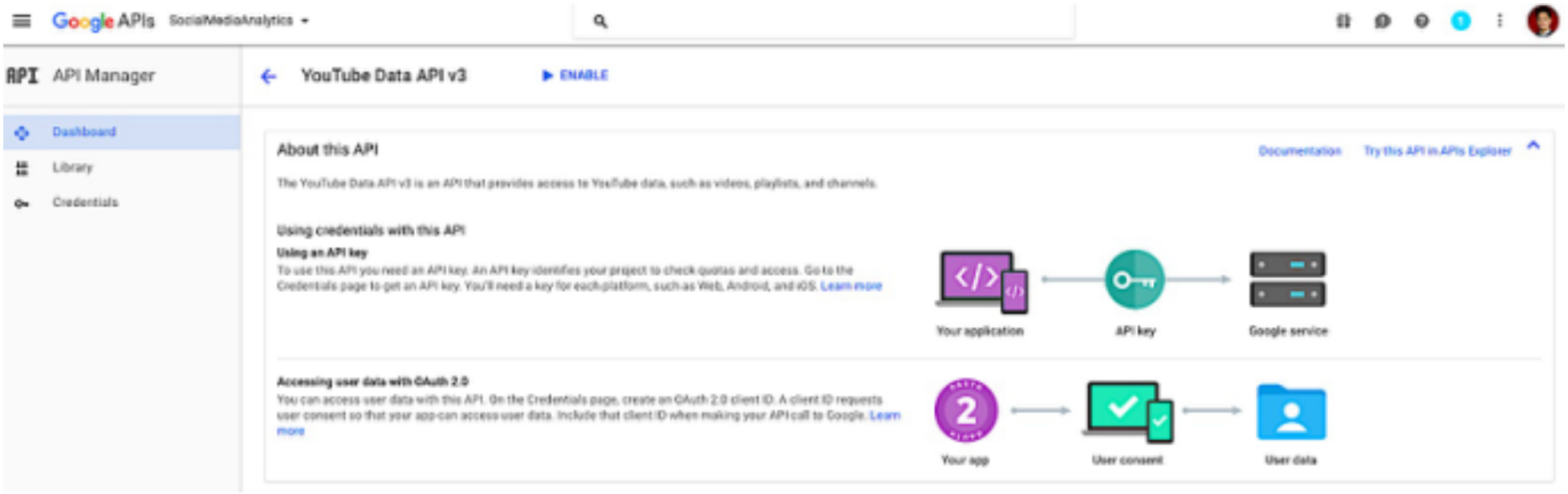
Google APIs Set-up

Created project - SocialMediaAnalytics



Google APIs Set-up

Enable ‘YouTube Data API v3’



Google APIs Set-up

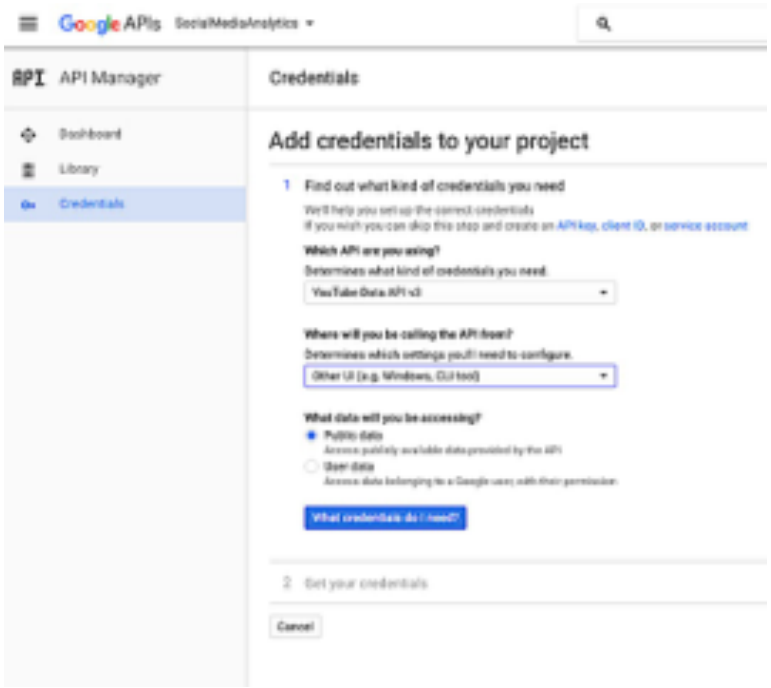
After YouTube Data API 3 is enabled, you are asked to create credentials for this API use.

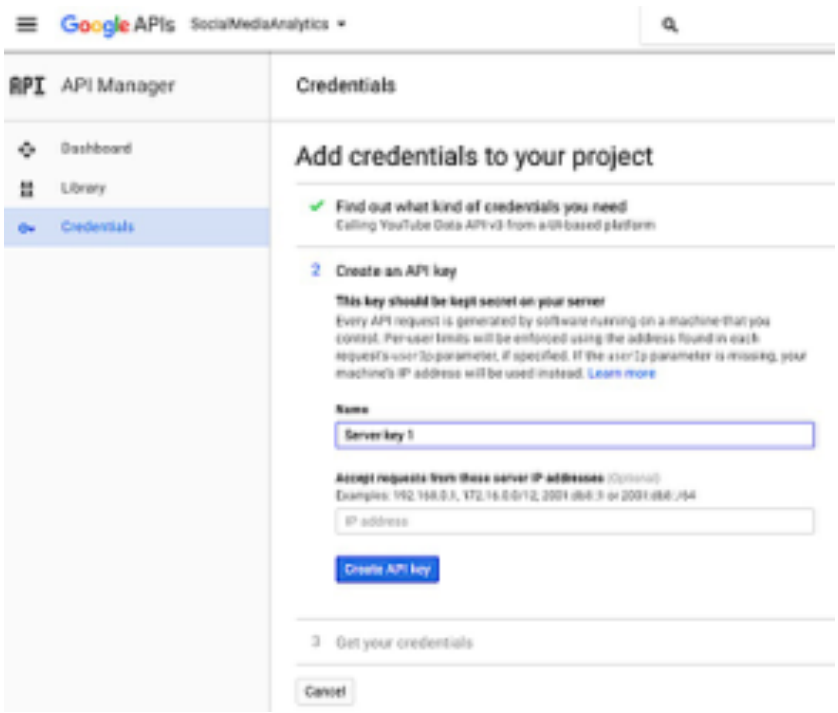
Click the ‘Create Credentials’ button.

Google APIs Set-up

You can configure credentials based on your purpose.

I chose the options as follows:





Google APIs Set-up

You have now the API key for your application. It looks like this:

YIzaSyCsABEu4ffovhN9WTk9mqMqewhPmO0LuRz

Sample Code

The script, ***youtube_search.py***, will retrieve YouTube content with a keyword it receives as an argument and show the results. The code will store video results - *title, videoId, viewCount, likeCount, dislikeCount, commentCount, and favoriteCount* - in ***video_result.csv*** file in the same directory.

Edit the script to add your API key at the line “DEVELOPER_KEY = ”.

youtube_search_keyword.py

Put your API key in the variable of DEVELOPER_KEY.

Run the script using command line (console) or through Spyder ‘Run configuration’:

In console

$ python ./youtube_search.py --q olympics

In Spyder

Set argument as ‘--q olympics’