Instructions to Extract YouTube Data

BDWY

Michael Friscia

Francisco Mendoza

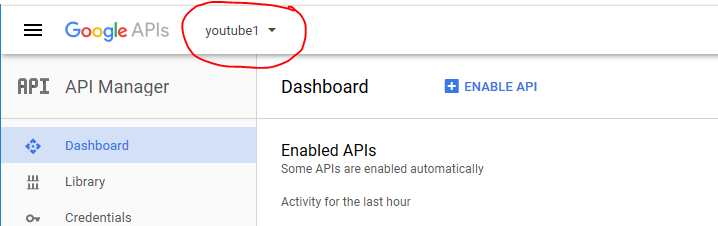
Yi Cai

Zheyu Tian

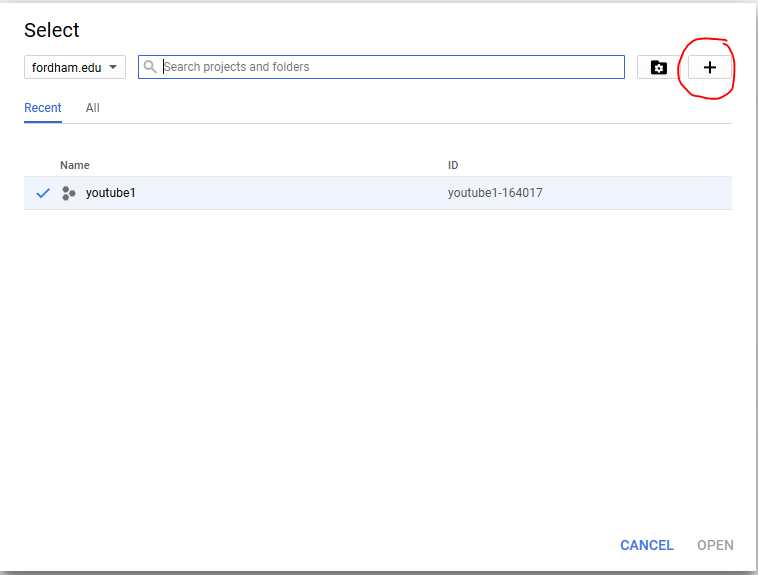
Professor Dobin Yim

4/25/17

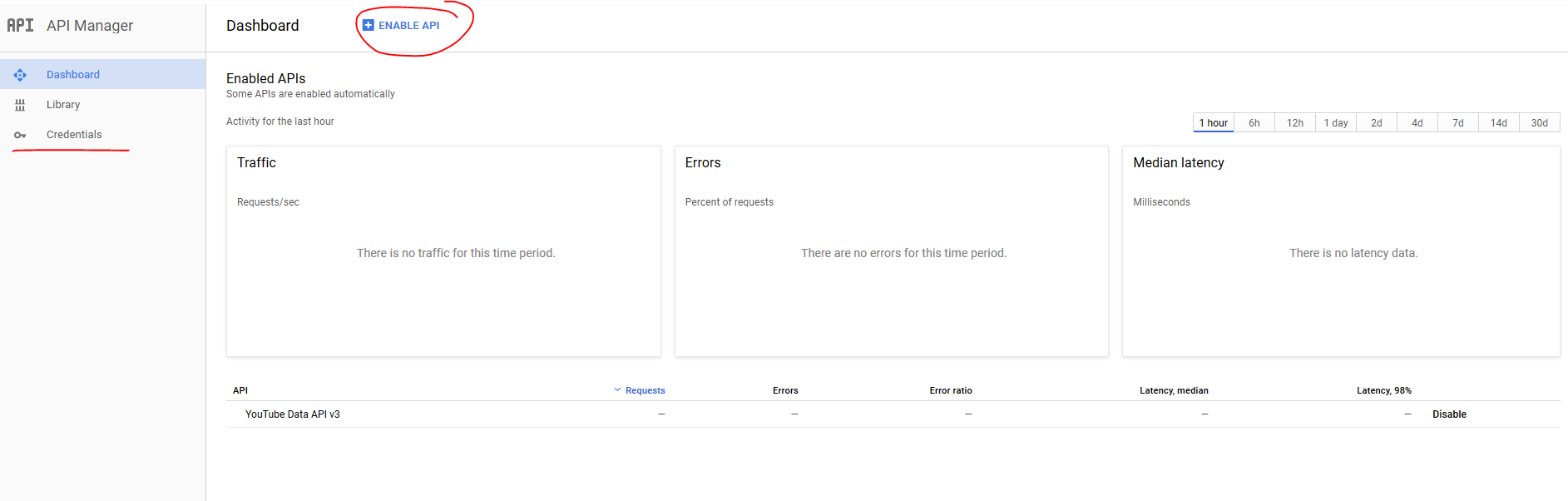
1. Go to console.developers.google.com and sign up for a trial account.
2. Click the project selector tool.



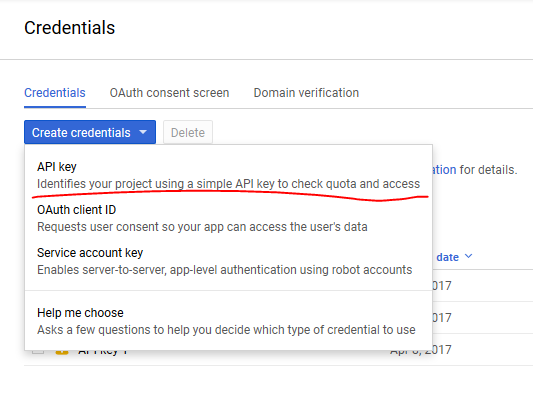
1. Create a new project.



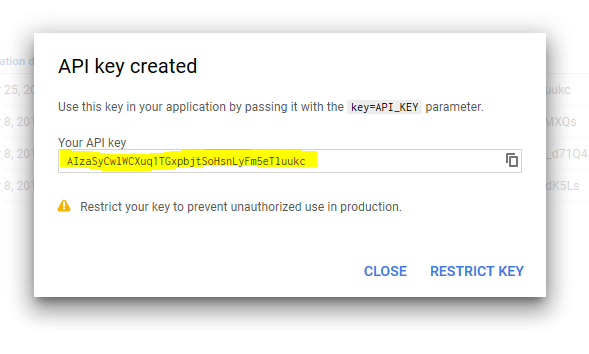
1. After creating a new project, go to the Dashboard, enable API, and then click Credentials on the side bar.



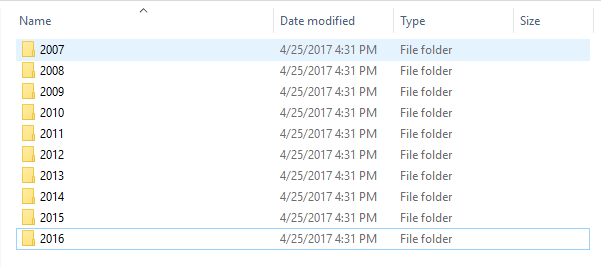
1. Create new credentials by clicking the blue create credentials button and select API Key.



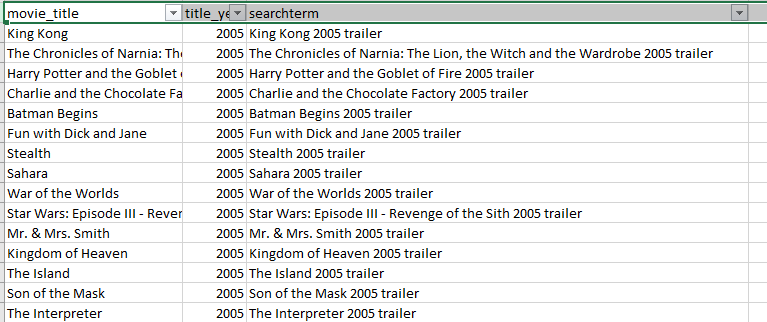
1. Copy the API Key and close.

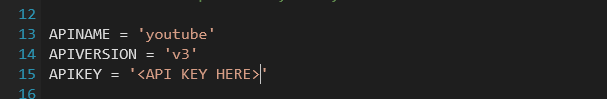


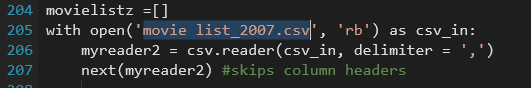
1. Extract the **YouTube Data Extraction Files** into a directory of your choice
2. Create different folders for years 2007 to 2016. This will prepare the environment for parallel instances of running the extraction code.



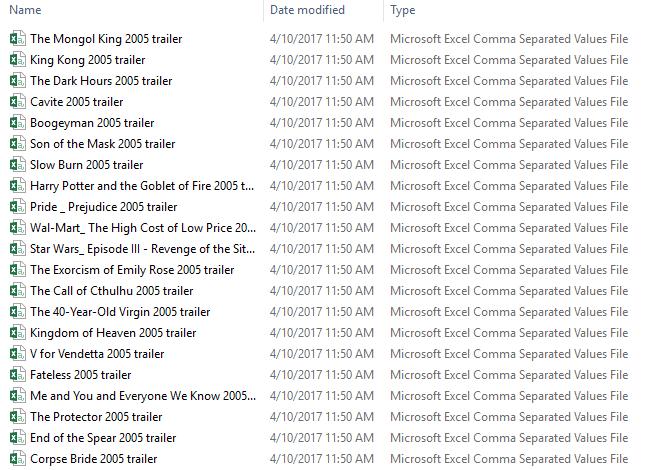
1. Open the **movie list.csv** file and separate it into smaller files, by year. Put each file into the corresponding year folder. File names should be in this format 🡪 **movie list\_2008.csv**



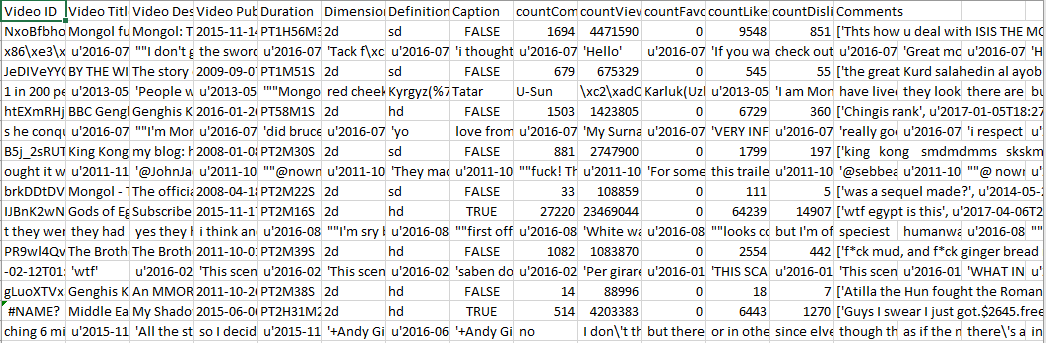
1. Open the tryallmovies.py file and paste your API key into the variable equation. 
2. Create a copy of the file tryallmovies.py in all of the year folders.
3. In each copy, go to line 205 and edit the file input name to the corresponding movie list file name.



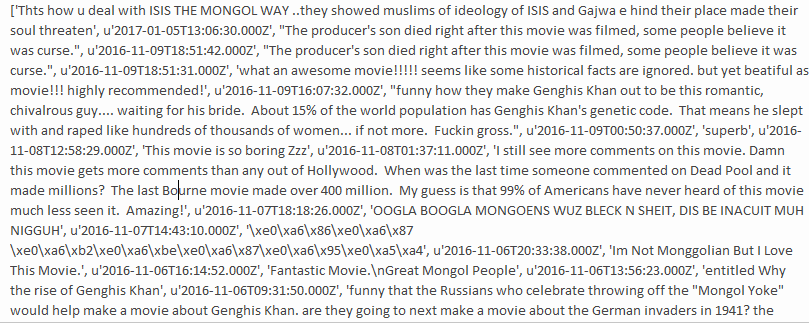
1. Go to CMD/Terminal, navigate to the directory, and run all python files.
2. The output will be something like this:



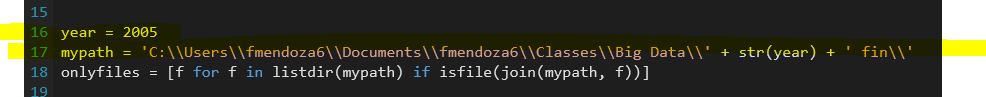
1. Each file will contain the youtube data information that we just extracted.



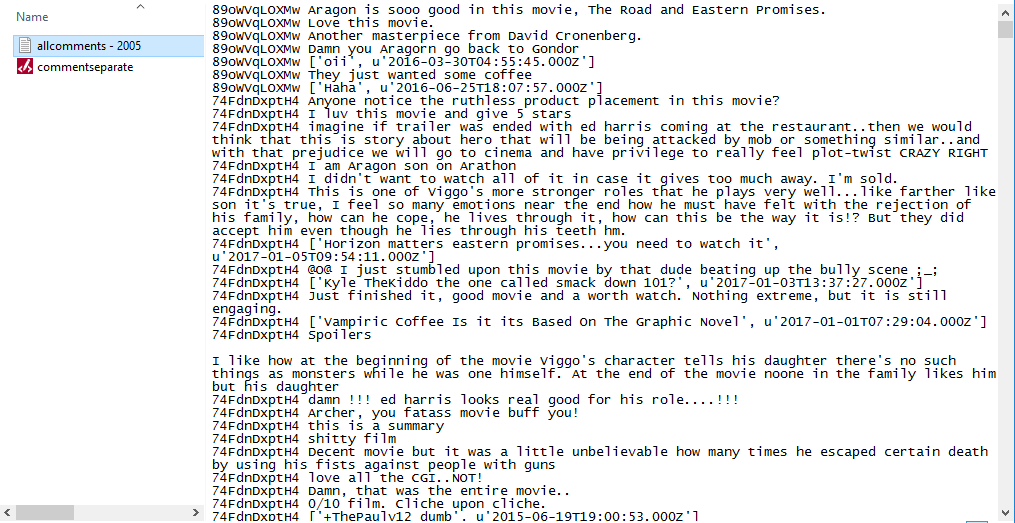
1. You will see that in the comments line, there is a long string. It contains all the comment threads and replies for each movie trailer of the movie. These will have to be extracted.
   1. Structure: [[Comment Thread1, date-time,[Comment reply1, date-time. Comment reply2,…], Comment Thread2,…]



1. Create a comments folder in the main directory. Inside it, create year folders again.
2. Copy the commentseparate.py file into each folder.
3. In the commentseparate.py file, there is a variable in line 16 called year. You will have to change the year depending on what folder it is in. You will also have to adjust the path into your current directory



1. This will make sure that your code opens the correct file.
2. The output file will be allcomments – 2005.txt, a tab delimited file. (e.g. alllcomments - <YEAR>.txt) It will contain all the comments and replies, with the corresponding video ID.



|  |  |  |
| --- | --- | --- |
| **YouTube Data Extraction** | | |
| Type | Filenames | Notes |
| Python Code | tryallmovies.py  tryallmovies2008.py  tryallmovies2009.py  tryallmovies2010.py  tryallmovies2011.py  tryallmovies2012.py  tryallmovies2013.py  tryallmovies2014.py  tryallmovies2015.py  tryallmovies2016.py | Youtube data extraction code. Separate each file into different folders from 2007 to 2016.  CSV files for each movie will be downloaded into each folder. (King Kong 2005 trailer.csv, etc.)  Separation is done so that system can download faster using parallel instances.  Ave. output file size <= 5MB |
| Reference file | movie list.csv | Reference file used in the tryallmovies.py extraction code. Separate file into smaller files, by year. E.g. movie list\_2007.csv, etc. |
| Python Code | commentseparate.py | Code that extracts all comments from trailer files (King Kong 2005 trailer.csv, etc.) and writes each comment to a tab delimited text file for sentiment analysis.  Adjust path to location of input files.  Ave. output file size <= 1GB |