

Collecting Data via WebAPIs in R

*Martin Schedlbauer, Ph.D., Yatish Jain*

This tutorial shows you how to use YouTube API to fetch data using R. Using YouTube API we can retrieve information like videos upload by any channel, number of views, likes, dislikes of any video. Data retrieved can be shaped using R making it conducive to analysis. To fetch any data using Youtube API user must register to Google console developer so that Google can keep track of the data retrieved and usually big data queries are charged. In this demonstration we will play with smaller dataset, which is free for public use. Like twitter API, Google also provides the option of oauth authentication but in this example we will see a way to bypass oauth authentication by generating browser key.

### Prerequiste:

* Have R installed directly in the C:// drive.
* Install RStudio once you have successfully installed R.

### Registering your Application on Google.

To run this script you need to generate your own “browserKey” by creating a new project in Google developer console and registering for youtube API.

Step 1: Login with your gmail ID on : <https://console.developers.google.com/project>

Step 2: Click on create project to create a new project for youtube API. Enter a name for your project like “Youtube data collection”. Once the project is all set click on the APIs and auth to expand the links on left side dashboard and click on APIs.

Click on first link under youtube APIs section named YouTube Data API and then click on enable API to use YouTube Data API v3.

Step 3: Next step is to generate browser key required to fetch any data from youtube api. Click on credentials link under the APIs and auth tab on left side dashboard. Under the public API access section click on create new key. Different keys can be generated based on how the data is fetched. We are going to fetch data through browser so let’s generate a browser key. Click on browser key, you can add referrers if you are fetching data from any other website of yours but here we will fetch data via R so leave this field blank and click on create. Copy this generated key for later use.

### Environment Setup :

Step 1: Change the working directory to Desktop using setwd command:

Mac users: setwd("/Users/Yatish/Desktop")

Windows: setwd("C:/Users/Yatish/Desktop")

Step 2: Load the following libraries after installing their packages.

install.packages(“RCurl”)

install.packages(“rjson”)

library(RCurl)

library(rjson)

RCurl library is used to scrape data from webpages. We will use function getURL to get the HTML elements in R variable.

rjson library is used to deal with json elements. We will use fromJSON function to convert the JSON string to list in R which is easy to manipulate.

Step 3: Set the browser key to the key generated in above steps:

Key<- “XXXX”

Step 4: Obtaining YouTube video id:

In the first example we will just fetch some simple statistics on any video to make sure our API connection is working fine.

Open <http://www.youtube.com> and browse any popular video for which you would like to fetch statistics.

After you are on video page, URL will look something like this: <https://www.youtube.com/watch?v=RgKAFK5djSk>

In this URL anything after v= is considered as video ID. We will need to use this video ID to pass argument to our stats function.

Step 5: Obtaining YouTube channel id:

As part of our second example we will fetch statistics of videos on a particular YouTube channel. To get a channel ID go to <http://www.youtube.com> and on the left side dashboard click on browse channels.

Using the search channel search box search for any channel. Click on the link for any channel the URL should look something like this: <https://www.youtube.com/channel/UCz6NJuz0ss3TxW7Fw4h_KIg>

In this URL anything after channel/ is a channel ID i.e. UCz6NJuz0ss3TxW7Fw4h\_KIg.

Note- sometimes you might get a URL like this:

<https://www.youtube.com/user/superherosachin> It just means that this is not a channel but a user. Click some other link to find channel.

### Source Code

Please locate the source code in the file "youTubeAPI.R". Here is the copy of the source code:

############################################################

## change the working directory to desktop to keep track of output file.

setwd("/Users/Yatish/Desktop")

getwd()

# Load required libraries

library(rjson)

library(RCurl)

#Generate your own key as per the instructions in document and paste it here to configure the API

key<- "XXXX"

#Funtion to check connection. This getStats function will fetch the statistics of any video given the video ID and key.

#Read instructions on how to get the video ID under the environment setup section

getStats <- function(id,key){

url=paste("https://www.googleapis.com/youtube/v3/videos?id=",id,"&key=",key,"&part=statistics,snippet",sep="")

raw.data <- getURL(url)

rd <- fromJSON(raw.data)

title<- rd$items[[1]]$snippet$title

channelTitle<- rd$items[[1]]$snippet$channelTitle

views<- rd$items[[1]]$statistics$viewCount

likes<- rd$items[[1]]$statistics$likeCount

dislikes<- rd$items[[1]]$statistics$dislikeCount

fav<- rd$items[[1]]$statistics$favoriteCount

comments<- rd$items[[1]]$statistics$commentCount

return(data.frame(title,channelTitle,views,likes,dislikes,fav,comments))

}

id<- ("4OIDdeGI7f8") #Change this video ID with some other

stats<-getStats(id,key)

stats

id<- ("QcIy9NiNbmo")

stats<-getStats(id,key)

stats

#getVideos function return the list of videos along with their statistics given the channelID and key.

# Read the instructions on how to get channelID under the environment setup section

getVideos<- function(channelID,key){

url=paste("https://www.googleapis.com/youtube/v3/search?key=",key,"&channelId=",channelID,"&part=snippet,id&order=date&maxResults=10",sep="")

raw.data <- getURL(url)

rd <- fromJSON(raw.data)

perPage<- rd$pageInfo$resultsPerPage

totalResults<-rd$pageInfo$totalResults

totalVideos<-min(perPage,totalResults)

stats<-c(NA,NA,NA,NA,NA,NA,NA)

for (i in 1:totalVideos){

kind<- rd$items[[i]]$id$kind

if(kind == "youtube#video"){

videoID<- rd$items[[i]]$id$videoId

stats<-rbind(stats,getStats(videoID,key))

}

else if(kind == "youtube#playlist"){

playlistID<- rd$items[[i]]$id$playlistId

url=paste("https://www.googleapis.com/youtube/v3/playlistItems?part=snippet%2CcontentDetails&maxResults=10&playlistId=", playlistID,"&key=",key,sep="")

raw.data <- getURL(url)

rd1 <- fromJSON(raw.data)

perPage<- rd1$pageInfo$resultsPerPage

totalResults<-rd1$pageInfo$totalResults

totalVideos<-min(perPage,totalResults)

for(i in 1:totalVideos){

videoID<-rd1$items[[i]]$contentDetails$videoId

stats<-rbind(stats,getStats(videoID,key))

}

}

}

return(stats)

}

channelID<-"UCANLZYMidaCbLQFWXBC95Jg" #change this channelID and run the function again

getVideos(channelID,key)

#getChannelsOrPlaylists function return the list of videos and their statistics associated with a keyword search on YouTube.

# When you search a keyword on youtube sometimes playlists also end up in search and we fetch the data from those playlists as well which might not be directly related to our search keyword, but it will fetch the data of similar searches.

getChannelsOrPlaylists<- function(search, key){

search<-URLencode(search)

url<-paste("https://www.googleapis.com/youtube/v3/search?q=",search,"&key=",key,"&type=channel&part=snippet&maxResults=50",sep="")

raw.data <- getURL(url)

rd <- fromJSON(raw.data)

perPage<- rd$pageInfo$resultsPerPage

totalResults<-rd$pageInfo$totalResults

totalChannels<- min(perPage,totalResults)

data<-NA

totalChannels<- min(perPage,totalResults)

data<-NA

for(i in 1:totalChannels){

channelID<- rd$items[[i]]$id$channelId

print(channelID)

if(!is.null(channelID)){

data<-rbind(data,getVideos(channelID,key))

}

}

data<- data[complete.cases(data),]

if(exists("data.csv"))

{

file.remove("data.csv")

}

write.csv(data,"data.csv")

}

search<-"taylor swift"

getChannelsOrPlaylists(search,key)