

```

# MyGov Gmail Contacts
# written by Bret Costain
# April 27, 2015
# Retrieve Users Address based on GoogleID
# Execute curl script to retrieve contacts relevant to User address
# Flatten nested JSON file
# Output CSV file
# Output Google+ Circles, Facebook List, Twitter List, and YouTube List
# NEXT STEP: CSV File imported in Gmail into "myGov" folder within Contacts
#
# BEGIN
# Set Working Directory
setwd("~/Desktop/MyGov")
# Install Packages
install.packages("RCurl")
library("RCurl")
install.packages("devtools")
library(devtools)
install.packages("rjson")
library("rjson")
install.packages("jsonlite")
library("jsonlite")
install.packages("foreach")
library("foreach")
install.packages("stringr")
library("stringr")
#
# FORMAT ADDRESS AS CURL EXPRESSION TO RETRIEVE USER ADDRESS
API_KEY = "AIzaSyA7KrZQYcgLz00c40bJL9G0f3IIsznpmro"
userGooglePlusID = "102823877180866545771"
expression1 = "https://www.googleapis.com/plus/v1/people/" # First part of getURL command BEFORE userGooglePlusID
expression2 = "?fields=placesLived&key=" # First part of getURL command AFTER userGooglePlusID BEFORE API_KEY
addressRequest = paste(expression1, userGooglePlusID, expression2, API_KEY, sep="")
addressTest = getURL(addressRequest)
#
# LOAD AS JSON ADDRESS AND CLEAN
json_address <- fromJSON(addressTest)
userAddress=json_address$placesLived$value
userAddressClean = str_replace_all(userAddress, ",", "") # REMOVE COMMAS
userAddressClean = str_replace_all(userAddress, " ", "%20") # REPLACE SPACES with %20
#
# RETRIEVE CONTACTS FROM GOOGLE CIVIC API
a = "https://www.googleapis.com/civicinfo/v2/representatives?key=AIzaSyA7KrZQYcgLz00c40bJL9G0f3IIsznpmro&address=" # First part of getURL command BEFORE clean address
c = paste(a, userAddressClean, sep = "")
z = getURL(c)
#
# CREATE LABELS
# Requires authorization
# POST https://www.googleapis.com/gmail/v1/users/userId/labels
# h = getURL ("https://www.googleapis.com/gmail/v1/users/102823877180866545771/labels
# ?fields=placesLived&key=AIzaSyA7KrZQYcgLz00c40bJL9G0f3IIsznpmro")
#
# Load JSON File
json_file <- z
json_data <- fromJSON(json_file)
#
# CLEAN NULL FROM EMAIL
json_data$officials$email[sapply(json_data$officials$email, is.null)] <- NA
json_data$officials$emails[sapply(json_data$officials$emails, is.null)] <- NA
# CLEAN LEVEL
json_data$offices$level[sapply(json_data$offices$level, is.null)] <- "Local" # Change NULL to Local
json_data$offices$level <- gsub("administrativeArea1", "State", json_data$offices$level) # Change administrativeArea1 to State
json_data$offices$level <- gsub("country", "Federal", json_data$offices$level) # Change country to Federal
#
# CREATE OFFICE RELATED VECTORS
officialInd=NULL
officeInd=NULL
officeName=NULL
officeLevel = NULL
q = json_data$offices$officialIndices
imax=length(q)
for (i in 1:imax )
{
  smax=length(q[[i]])
  for (s in 1:smax )
  {
    officialInd=c(officialInd,q[[i]][s])
    officeInd=c(officeInd,i)
    officeName=c(officeName,json_data$offices$name[i])
    officeLevel=c(officeLevel,json_data$offices$level[i])
  }
}
#
# CREATE OFFICIAL VECTORS
officialLongName = (json_data$officials$name) # CREATE OFFICIAL NAME VECTOR
e = (json_data$officials$emails) # CREATE OFFICIAL EMAIL VECTOR
f = as.character(e) # CONVERT EMAIL to CHARACTER
ph = (json_data$officials$phone)
ph2 = sapply(ph, "[", 1) #Select first phone number
circle = as(officeLevel,"character")
photo = json_data$officials$photoUrl
officialFirstName=word(officialLongName,1)
officialLastName=word(officialLongName,-1)
#
# CREATE MASTER DATE FRAME FOR CONTACTS
master = data.frame(officialLongName, officeName, f, ph2, circle, photo, officialInd, officeInd)
colnames(master) <- c("Name", "Organization", "Email", "phone", "Circle", "Photo", "officialInd", "officeInd")
write.csv(master, file = "MyGov.csv") # CREATE CSV File
#
# CREATE CHANNELS LISTS
channels=json_data$officials$channel
channels[sapply(channels, is.null)] <- "NA" # Change NULL to Local
# SET NULLS
channelID=NULL
channelType=NULL
channelOffName=NULL
channelCircle=NULL
#

```

```

imax=length(channels)
for (i in 1:imax )
{
smax=length(channels[[i]][2][[1]])
for (s in 1:smax )
{
channelType<-c(channelType,channels[[i]][1][[1]][[s]]) #extract Channel Type
channelID<-c(channelID,channels[[i]][2][[1]][[s]]) # extract Channel ID
channelOffName<-c(channelOffName,json_data$officials$name[i]) # Log Name for each row
channelCircle<-c(channelCircle,circle[i]) # Log Circle for each row
}
}
#
# CREATE MASTER DATAFRAME FOR CHANNELS
channelDF<-data.frame(channelOffName,channelType,channelID,channelCircle)
# GOOGLE+ LIST
GP = which(channelDF$channelType=="GooglePlus")
googleList = channelDF[GP,]
googleList2 = data.frame(googleList$channelOffName,googleList$channelID,googleList$channelCircle)
colnames(googleList2) <- c("Name", "GooglePlus", "Circle")
# FACEBOOK LIST
FB = which(channelDF$channelType=="Facebook")
facebookList = channelDF[FB,]
facebookList2 = data.frame(facebookList$channelOffName, facebookList$channelID)
colnames(facebookList2) <- c("Name", "Facebook")
# TWITTER LIST
TW = which(channelDF$channelType=="Twitter")
twitterList = channelDF[TW,]
twitterList2 = data.frame(twitterList$channelOffName,twitterList$channelID)
colnames(twitterList2) <- c("Name", "Twitter")
# YOUTUBE LIST
YT = which(channelDF$channelType=="YouTube")
youtubelist = channelDF[YT,]
youtubelist2 = data.frame(youtubelist$channelOffName,youtubelist$channelID)
colnames(youtubelist2) <- c("Name", "Youtube")
# OUTPUT FILES
write.csv(googleList2, file = "googleList.csv") # Create Google+ file
write.csv(facebookList2, file = "facebookList.csv") # Create Facebook file
write.csv(twitterList2, file = "twitterList.csv") # Create Twitter file
write.csv(youtubelist2, file = "youtubelist.csv") # Create YouTube file
#
# END

```