

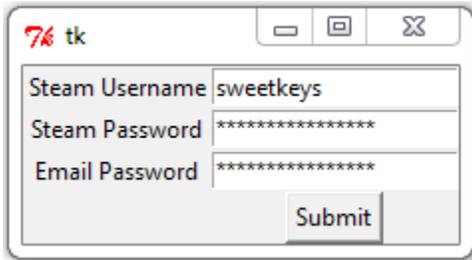
Steam Web Browser Automation Script

Steam game trading is a big market on Steam. It is a tedious process and required about 10-15 hours of work a week for those who wish to trade their games. The process requires one to repeatedly interact with the browser through forms and extract links from their e-mails. Thus, in the summer of 2014 I decided to write a Python script to automate this process by parsing and interacting with my e-mail and browser. After writing this script, I open sourced it and sent it to many game traders to assist their game trading business and save them hundreds of hours of manual labor.

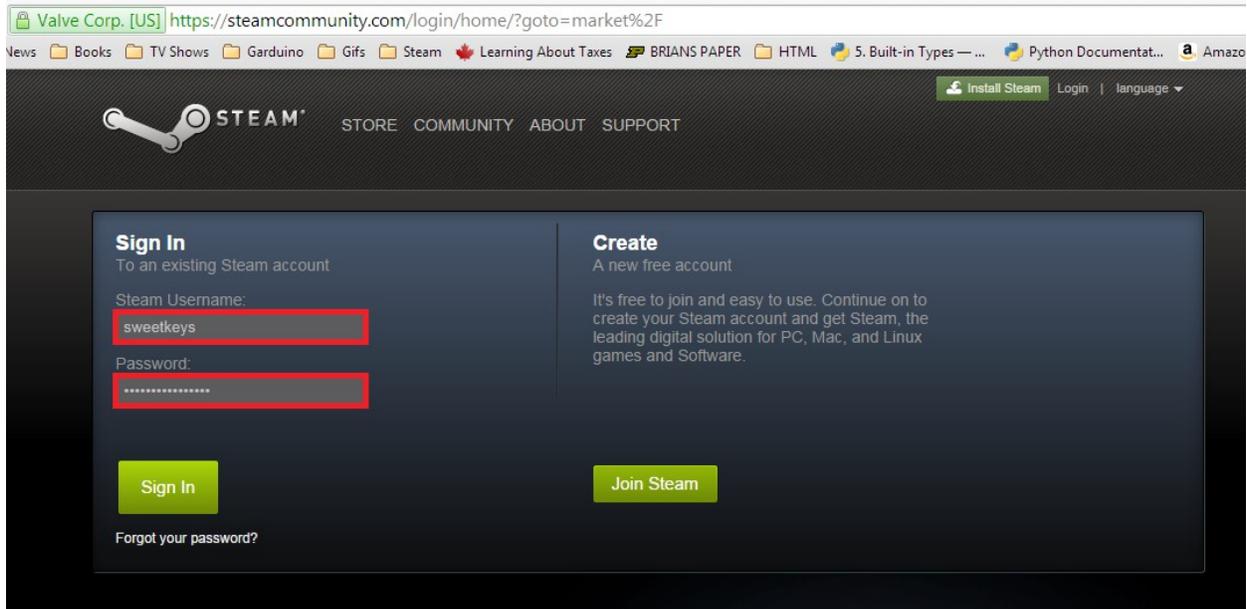
This project has allowed me to gain a significant amount of software experience. To parse and communicate with the webpage I used a Selenium WebDriver. I looked for the "id" attributes of the forms and buttons in the webpages HTML. After retrieving these attributes, I used Selenium to interact with these elements. For username and password input, I used the Tkinter GUI package to hide the user's password input. Finally, I used the imaplib module to login and parse my e-mails. This allowed me to extract the authentication codes required to log into Steam and also the gift links for the game.

Because it is difficult to explain exactly what my code does, I have created this portfolio. The following is a set of screenshots explaining exactly what my script does followed by the source code.

1. Display GUI to input username and passwords using the Tkinter GUI package



2. Interact with browser HTML to input username and password



My code uses the Selenium WebDriver to access different forms based on the id of the form in the webpage's HTML.

3. Parses e-mail for authentication code to log in

Gmail ▾ 1-50 of 802

Primary Social 1 new Google+ Promotions +

COMPOSE

Inbox (1)
Starred
Sent Mail
Drafts
Steam
Steam Access (1)

Steam Support Steam Access Your Steam account: Access from new computer 11:14 pm

Steam Support Steam Access Your Steam account: Access from new computer 11:13 pm

Steam Store Steam You've received a gift copy of the game Insurgency on Steam - Trouble viewing this message? Click here. 1 Sep 6

Steam Support Steam Access Your Steam account: Access from new computer Sep 6

Your Steam account: Access from new computer Inbox x Steam Access x

Steam Support <noreply@steampowered.com> 11:06 PM (9 minutes ago) ☆

to me ▾



Dear swagkeys,

We've received a request to access your Steam account from a new browser located at IP address: **24.84.60.5**
Our records show this IP address is in **VANCOUVER BRITISH COLUMBIA CA**

If that location does not reasonably match your location or the location of your Internet Service Provider, then follow this link for more information:

[Verify Login Location](#)

If the information provided does not match then do not proceed, close the browser window, and change your password immediately.

If you did not attempt this action, please change your password immediately or follow this link to the Steam Support site to lock your account and submit a request for assistance:

[Link to lock account](#)

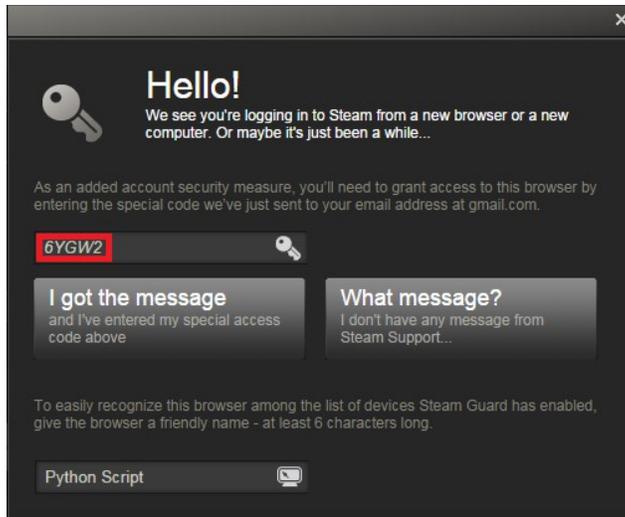
To complete this process, enter the following special access code into the authorization dialog before trying to log in again:

6YGW2 ← Authentication Code

Thanks for helping us maintain the security of your account.

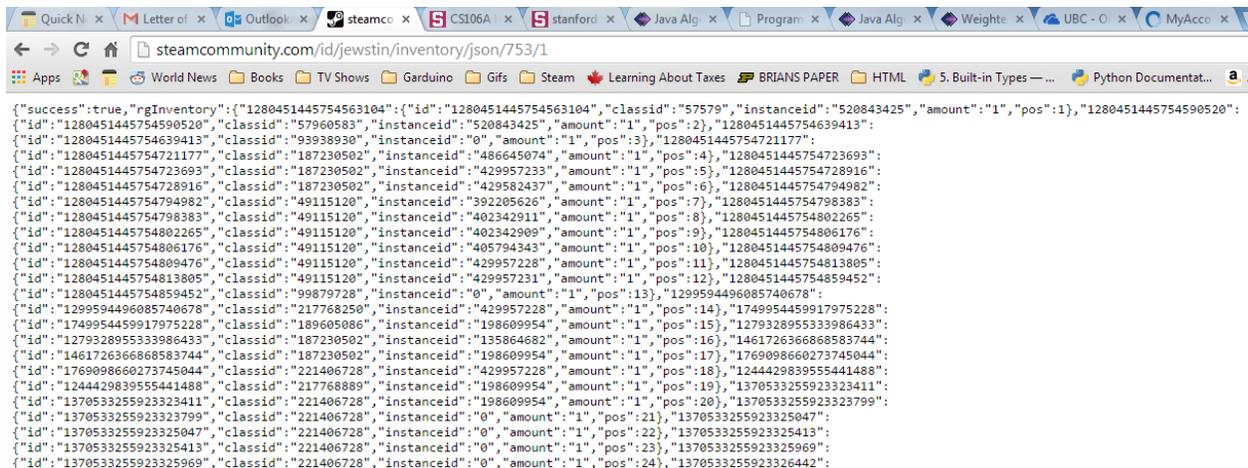
The Steam Support Team
<https://support.steampowered.com>

Upon attempting to log in, Steam will send an e-mail with an authentication code as a way to guard against account theft. My script opens this e-mail and parses it for the authentication code.



It then goes back to the webpage and inputs the authentication code.

4. Accesses JSON data and parses it on the webpage with regular expressions



Using the Selenium WebDriver, I can access the games in my inventory and grab their JSON formatted data. I can parse this to find out which games I want to send.

5. Extract the JSON data to access each game and send it to my e-mail

orp. [US] <https://store.steampowered.com/checkout/sendgift/249143227632764759> ← From JSON data

ks TV Shows Garduino Gifs Steam Learning About Taxes BRIANS PAPER HTML 5. Built-in Types ... Python Documentat...

Gift Delivery ▸ Gift Note

INSURGENCY Insurgency STEAMPLAY

YOUR GIFT'S ARRIVAL
Your gift will be delivered digitally using the method you choose.

Choose a delivery method:

Email my gift
Send my friend a gift email message, including all the info they'll need to get up and running on Steam.
Recipient's email address
 ← Generator function generates different e-mail combinations

Send my gift directly through Steam
Send my friend a gift message directly to their Steam account.
Note: This option also sends an email message, to be sure your friend sees your gift as soon as possible.

NEXT > CANCEL

Gift Delivery ▸ **Gift Note**

INSURGENCY Insurgency Sending to: Email address **sweetkeyssteam@gmail.com** STEAMPLAY

YOUR GIFT'S ARRIVAL
Your gift will be delivered digitally using the method you choose.

Personalize your gift with a little note

Recipient's first name

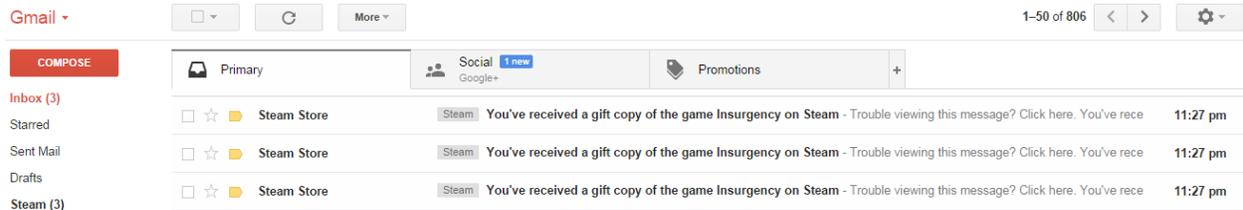
Gift message (160 character limit, 135 left)

Your sentiment

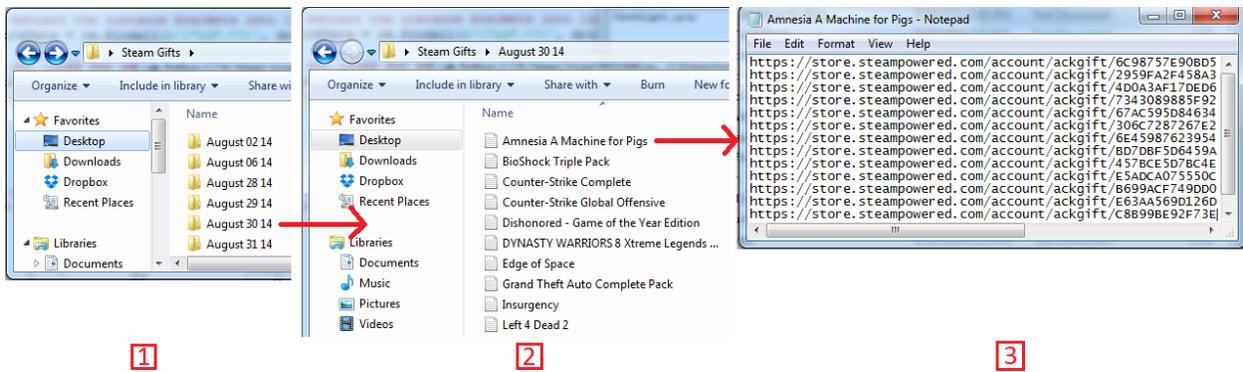
Your signature

From the JSON data, I extracted links for all the games and looped through my game inventory until I sent all the games to my e-mail.

6. Parse the game gift links in my e-mail and extract the links into text files.



After the games are sent to my e-mail I can extract the links by parsing the contents of the e-mails.



The parsed game links are organized in folders by the date in which they were extracted as seen in section 1 of the image above. They are also organized by the name of the game as seen in section 2. Finally, the links can be found in the text file as seen in section 3.

Source Code to Interact with Browser

```
import sys
import getpass
import imaplib
import os
import time
import re
import itertools

from Tkinter import *
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC

# class for gui window to get user and pass
class gui:
    def __init__(self, master):
        self.master = master

        # labels for each entry
        Label(self.master, text="Steam Username").grid(row=0)
        Label(self.master, text="Steam Password").grid(row=1)
        Label(self.master, text="Email Password").grid(row=2)

        # button widget
        self.steamUserW = Entry(self.master)
        self.steamPassW = Entry(self.master, show="*")
        self.emailPassW = Entry(self.master, show="*")
        self.submit = Button(self.master, text="Submit", command=self.assign)

        # bind the ENTER key to callback function
        self.emailPassW.bind("<Return>", self.assign)
        self.emailPassW.bind("<KP_Enter>", self.assign)

        # space out the widgets
        self.steamUserW.grid(row=0, column=1)
        self.steamPassW.grid(row=1, column=1)
        self.emailPassW.grid(row=2, column=1)
        self.submit.grid(row=3, column=1)

        # grabs the values in the entry boxes and assigns them to variable
        def assign(self, *args):
            self.steamUser = self.steamUserW.get()
            self.steamPass = self.steamPassW.get()
            self.emailPass = self.emailPassW.get()
            self.close()

        # closes GUI window
```

```

    def close(self):
        self.master.destroy()

root = Tk()
userGui = gui(root)
root.mainloop()
steamUser = userGui.steamUser
steamPass = userGui.steamPass
emailPass = userGui.emailPass

# open up log-in screen
chromedriver = "/Users/Justin/Desktop/Steam Gifts/Programs/chromedriver"
os.environ["webdriver.chrome.driver"] = chromedriver
driver = webdriver.Chrome(chromedriver)
driver.get("https://steamcommunity.com/login/home/")

# enter username and password
username = driver.find_element_by_id("steamAccountName")
username.send_keys(steamUser)
password = driver.find_element_by_id("steamPassword")
password.send_keys(steamPass, Keys.TAB, Keys.ENTER)

# process mailbox to get special code
# get Steam access code from e-mail
def process_mailbox():
    M = imaplib.IMAP4_SSL('imap.gmail.com')
    M.login('sweetkeys.steam@gmail.com', emailPass)
    rv, data = M.select("Steam Access")
    rv, data = M.search(None, '(UNSEEN)')

    ids = data[0] # data is a list
    id_list = ids.split() # ids is a space seperated string

    # parses email, creates folder, creates text file and appends steam gift
    link to text file
    for i in range(len(id_list)):
        # get e-mail body text
        result, data = M.fetch(id_list[i], "(RFC822)")

        # get the link for the Steam gift
        startPos = data[0][1].find("<h3>")
        M.close()
        M.logout()
        return data[0][1][startPos+4:startPos+9]

specialCode = None
while specialCode == None:
    specialCode = process_mailbox() # ... do something with emails, see below
    ...

# input special code
specialCodeForm = driver.find_element_by_id("authcode")
specialCodeForm.send_keys(specialCode)

```

```

# input computer name
computerNameForm = driver.find_element_by_id("friendlyname")
computerNameForm.send_keys("Firefox Steam Gifts", Keys.TAB, Keys.ENTER)

# get JSON data
wait = WebDriverWait(driver,10)
element = wait.until(EC.visibility_of_element_located((By.ID,
"success_continue_btn")))
driver.get("http://steamcommunity.com/id/" + steamUser +
"/inventory/json/753/1")

# extract the instance brackets into linkData
linkData = re.findall(r'{"id".*?}', driver.page_source)

# get links for each gift
linkList = []
for links in linkData:
    instanceid = re.findall(r'"instanceid": "(\\d*)"', links)
    if instanceid[0] == '0':
        id = re.findall(r'"id": "(\\d*)"', links)
        linkList.append("http://store.steampowered.com/checkout/sendgift/" +
id[0])

# generator function to get different combinations of my e-mail
# sweetkeyssteam@gmail.com and s.weetkeyssteam@gmail.com are two examples of
# e-mail combinations that will be generated
def emailComboGen():
    sweetList = ['s', 'w', 'e', 'e', 't', 'k', 'e', 'y', 's', 's', 't', 'e', 'a', 'm']
    sweetNumbers = []
    i = 1
    j = 0
    k = 0

    while i < len(sweetList):
        sweetNumbers.append(i)
        i = i + 1

    while j < len(sweetList):
        emailCombos = itertools.combinations(sweetNumbers, j)
        for combo in emailCombos:
            sweetListCopy = sweetList[:]
            k = 0
            for index in combo:
                sweetListCopy.insert(index + k, '.')
                k = k + 1
            yield ''.join(sweetListCopy) + "@gmail.com"
        j = j + 1

# create generator for e-mails
email = emailComboGen()

time.sleep(5)

# go through and fill in forms for each steam gift
for link in linkList:
    # go to email steam gift link
    driver.get(link)

```

```

# fill in e-mail
emailGift = wait.until(EC.presence_of_element_located((By.ID,
"email_input")))
emailGift.send_keys(next(email), Keys.TAB, Keys.ENTER)
# fill in personalized message forms
name = driver.find_element_by_id("gift_recipient_name")
name.send_keys("Sir/Madam")
message = driver.find_element_by_id("gift_message_text")
message.send_keys("Have fun with the game, be sure to leave positive
feedback :)")
signature = driver.find_element_by_id("gift_signature")
signature.send_keys("Justin", Keys.TAB, Keys.TAB, Keys.ENTER)

# used to parse e-mail for game gift links and organize into folders
def process_mailbox(M):
    rv, data = M.search(None, '(UNSEEN)')

    ids = data[0] # data is a list
    id_list = ids.split() # ids is a space separated string

    # parses email, creates folder, creates text file and appends steam gift
link to text file
    for i in range(len(id_list)):
        # get e-mail body text
        result, data = M.fetch(id_list[i], "(RFC822)")

        # get the name of the game
startPosSubject = data[0][1].find("Subject: You've received a gift
copy of the game ")
endPosSubject = data[0][1].find("on Steam", startPosSubject)
gameName = data[0][1][startPosSubject + 49:endPosSubject - 1]
        # remove colons
        if ":" in gameName:
            indexOfColon = gameName.find(":")
            gameName = gameName[0:indexOfColon] + gameName[indexOfColon + 1:]

        # create folder for the day if it does not exist
folderName = time.strftime('%B %d %y')
folder = 'C:\\Users\\Justin\\Desktop\\Steam Gifts\\' + folderName +

'\\'

        if not os.path.exists(folder):
            os.mkdir(folder)

        # create a text file named after the game
f = open(folder + gameName + '.txt.', 'a')

        # get the link for the Steam gift
startPosGift = data[0]
[1].find("https://store.steampowered.com/account")
endPosGift = data[0][1].find(".com", startPosGift + 30)
steamGift = data[0][1][startPosGift:endPosGift + 4]

        # write link to the text file
f.write(steamGift + '\n')

f.close()

```

```
M = imaplib.IMAP4_SSL('imap.gmail.com')
M.login('sweetkeys.steam@gmail.com', emailPass)

rv, data = M.select("Steam") # select mailbox label
if rv == 'OK':
    print "Processing mailbox...\n"
    process_mailbox(M) # ... do something with emails, see below ...
    M.close() # close mailbox Steam
M.logout()
```