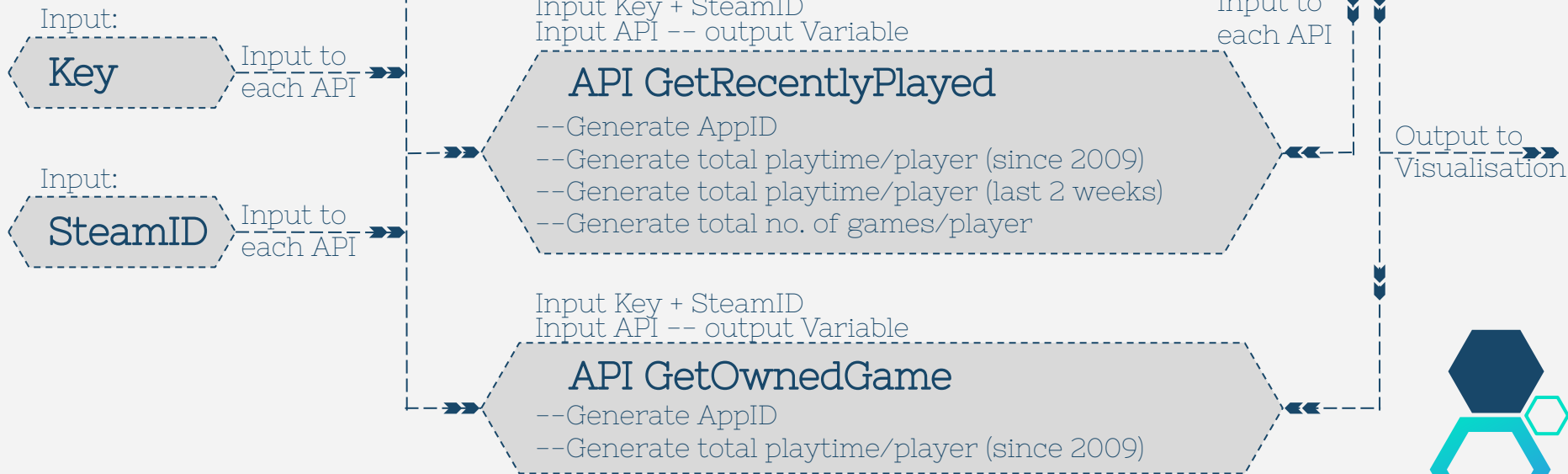


Content

- How to use the API – P.2
- How to get the API key – P.3
- How to get Steam Community ID --- Our approaches, how steam ID can be generated – P.4 – 8
- Using API key and Steam ID to find account information (Program structure, json format, ...) – P.9
- Reducing number of request (GetPlayerSummeries) – P.10



Project Framework -- Data Collection



- Key is a passcode to access API and Steam Games
e.g. 72797CA67785C46C4DDB70C6F4C295D3

- How to get API key:

https://support.steampowered.com/kb_article.php?ref=3330-IAGK

“Steam prevents some accounts from accessing a certain community and social features that are commonly used for spamming, phishing, and other abuse.”

(i) spend at least \$5.00 USD in Steam

(ii) install Steam to show that you are an active user

- Challenge: Only allow 100,000 Requests per day





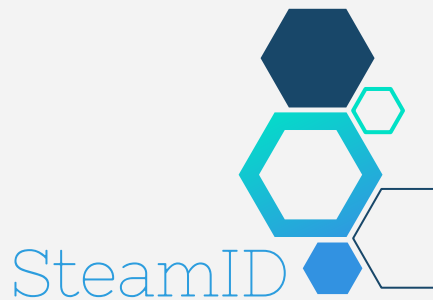
- A SteamID is a unique identifier used to identify a Steam account.

- There is a few ways to present the SteamID, including simple format (e.g. STEAM_X:Y:Z), 64-bit format and Steam Community ID in 17-bit format (e.g. 76561197960265728)

→ See P.7 for detail

SteamID

Input to
each API



Get Steam Community ID (i.e. the 17-digit steam ID)

1. Approach 1 --- generate random account numbers

→ by observation, valid steam IDs usually have a pattern (e.g.

76561198XXXXXXXXXX)

→ “76561198” + randomly generate 9-digit numbers

- Challenge --- low efficiency (below numbers tbc)

(i) Only 1% of SteamID generated is valid

(ii) Time required to select valid id (~1 valid ID/ second?)

2. Approach 2 --- generate the ID from its simple format (i.e.

STEAM_1:1:66138017)

(See next page for details)

76561198060265729
76561198060265731
76561198060265735
76561198060265793
76561198060265809
76561198060265813
76561198060265821
76561198060265827
76561198060265849
76561198060265865
76561198060265905
76561198060265907
76561198060265913
76561198060265931
76561198060265947
76561198060265953
76561198060265965
76561198060265967

Transforming SteamID to 17-digit number

Steam
Community ID

64-bit data
structure

Simple format

76561198092541763

0000000100010000000000000000000001

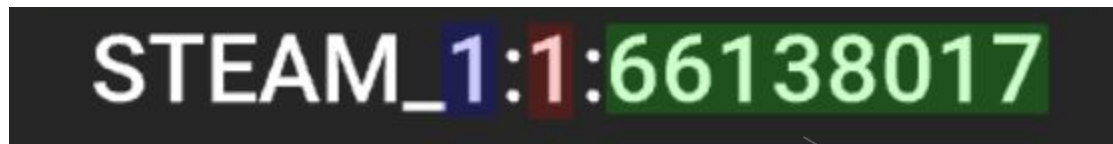
00000111111000100101111101000011

STEAM_1:1:66138017

Description	Format	Example
Simple format	SteamID	STEAM_1:1:66138017
64-bit data structure for computer programs	Binary signed 2's complement	000000010001000000000000000000000100001111110001001011111010000101
Steam Community ID for use on the Steam Community website	Decimal number	76561198092541763

Meaning of the simple SteamID

STEAM_1:1:66138017



Universes Available for Steam Accounts

--- “1” means it is a public account

→ If the account type is “public”, we can read user’s information.

→ So we default this value to be ‘1’

Account Number (usually 8 digits and consecutively given to users))

→ we can generate this number consecutively and get the Steam Community ID (see the next page)

Account Types --- “1” means individual account

- given URL Path

- SteamID64 Identifier for individual account

→ So we default this value to be ‘1’

(Should I explain it in detail?)

function - convert
simple Dteam ID into
Steam Community ID

Select 8-dig simple
steam ids consecutively

White into csv file

```
# find steamIDs
def steamid_to_64bit(steamid):
    steam64id = 76561197960265728

    id_split = steamid.split(":")
    steam64id += int(id_split[2]) * 2
    if id_split[1] == "1":
        steam64id += 1
    return steam64id

Id_list = []
multi_Id = []
for num in range(90000000, 90050000):
    steam_Id = "STEAM_1:1:" + str(num)
    Id_list.append(steam_Id)

# generate 64-bit steamID list and csv file
steamId_64bit = []
with open("steamid.csv", "w") as f:
    for i in Id_list:
        steamId_64bit.append(str(steamid_to_64bit(i)))
        steamId = str(steamid_to_64bit(i)) + ",\n"
        f.write(steamId)
```

*Source:

<https://stackoverflow.com/questions/36463687/how-can-i-get-a-steamid-64-from-a-steamid-in-python>

Code :

```
##Create a function to get coutrycode given steamId
def GetPlayerSummaries(steamId):
    #steamids string
    try:
        steamId = str(steamId)
        params = {
            "steamids" : steamId,
            "key" : "72797CA67785C46C4DDB70C6F4C295D3"
        }
        url = requests.get("https://api.steampowered.com/ISteamUser/GetPlayerSummaries/v2/" ,params=params)
        # print(url)
        data = json.loads(url.text)
        return data["response"]["players"][0]["loccountrycode"]
    except:
        return np.nan
```

API

steamID

key

Return data

Use Steam ID and API key to **get user's information**

- GetPlayerSummaries: 100 IDs per request

GetPlayerSummaries

```
GET https://api.steampowered.com/ISteamUser/GetPlayerSummaries/v2/
```

Name	Type	Required	Description
key	string	✓	Steamworks Web API user authentication key.
steamids	string	✓	Comma-delimited list of SteamIDs (max: 100)

```

import numpy as np
with open("steamid.txt", "r") as f:
    multi_id = f.read().split(",")

id_pub = []
def GetPlayerSummaries(multi_id):
    with open("id_location.csv", "w") as f:
        steamId = ""
        for i in multi_id:
            if steamId != "":
                steamId = steamId + "," + i
            else:
                steamId = i

        params = {
            "steamids" : steamId
        }

        url = requests.get("https://api.steampowered.com/ISteamUser/GetPlayerSummaries/v2/?key=81C")
        d = json.loads(url.text)

        for i in range(len(multi_id) - 1):
            try:
                if d['response']['players'][i]['loccountrycode']:
                    x = multi_id[i] + "," + d['response']['players'][i]['loccountrycode'] + "\n"
                    id_pub.append(x)
            except:
                d['response']['players'][i]['loccountrycode'] = np.nan

        for i in range(len(id_pub)):
            f.write(id_pub[i])

```

```

for i in range(0, 500):
    GetPlayerSummaries(multi_id[(0+100*i):(100+100*i)])

```

add for loop to input 100 steam ids every time