STEAM DOCUMENTATION:

https://developer.valvesoftware.com/wiki/Steam_Web_API

APPS IDS LIST

https://steamdb.info/apps/

Exemples:
. 581320 (Insurgency)
. 252490 (Rust)

STEAM API KEY

389EC943738900A510BF540217AFB042

STEAM ID

https://steamidfinder.com

- 76561197974733729: Slickity

STEAM API ENDDPOINTS:

GetPlayerSummaries (v0002)

Example URL: http://api.steampowered.com/ISteamUser/GetPlayerSummaries/v0002/?key=389EC943738900A510BF540217AFB042&steamids=76561197974733729 (This will show Slickity profile information.)
Returns basic profile information for a list of 64-bit Steam IDs.
Arguments

- steamids (Comma-delimited list of 64 bit Steam IDs to return profile information for. Up to 100 Steam IDs can be requested.)
- format (Output format. json (default), xml or vdf.1)

Public Data

- steamid
  - 64bit SteamID of the user
- personaname
  - The player's persona name (display name)
- profileurl
  - The full URL of the player's Steam Community profile.
- avatar
  - The full URL of the player's 32x32px avatar. If the user hasn't configured an avatar, this will be the default ? avatar.
- avatarmedium
  - The full URL of the player's 64x64px avatar. If the user hasn't configured an avatar, this will be the default ? avatar.
- avatarfull
  - The full URL of the player's 184x184px avatar. If the user hasn't configured an avatar, this will be the default ? avatar.
- personastate

  - The user's current status. 0 - Offline, 1 - Online, 2 - Busy, 3 - Away, 4 - Snooze, 5 - looking to trade, 6 - looking to play. If the player's profile is private, this will always be "0", except if the user has set their status to looking to trade or looking to play, because a bug makes those status appear even if the profile is private.
- communityvisibilitystate
  - This represents whether the profile is visible or not, and if it is visible, why you are allowed to see it. Note that because this WebAPI does not use authentication, there are only two possible values returned: 1 - the profile is not visible to you (Private, Friends Only, etc), 3 - the profile is "Public", and the data is visible. Mike Blaszczak's post on Steam forums says, "The community visibility state this API returns is different than the privacy state. It's the effective visibility state from the account making the request to the account being viewed given the requesting account's relationship to the viewed account."
- profilestate
  - If set, indicates the user has a community profile configured (will be set to '1')
- lastlogoff
  - The last time the user was online, in unix time.
- commentpermission
  - If set, indicates the profile allows public comments.

Private Data

- realname
  - The player's "Real Name", if they have set it.
- primaryclanid
  - The player's primary group, as configured in their Steam Community profile.
- timecreated
  - The time the player's account was created.
- gameid
  - If the user is currently in-game, this value will be returned and set to the gameid of that game.
- gameserverip
  - The ip and port of the game server the user is currently playing on, if they are playing on-line in a game using Steam matchmaking. Otherwise will be set to "0.0.0.0:0".
- gameextrainfo
  - If the user is currently in-game, this will be the name of the game they are playing. This may be the name of a non-Steam game shortcut.
- cityid
  - This value will be removed in a future update (see loccityid)
- loccountrycode
  - If set on the user's Steam Community profile, The user's country of residence, 2-character ISO country code
- locstatecode
  - If set on the user's Steam Community profile, The user's state of residence
- loccityid
  - An internal code indicating the user's city of residence. A future update will provide this data in a more useful way.
  - steam_location gem/package makes player location data readable for output.

Sample (JSON):

```
{
    "response":{
            "players":[
                {
                    "steamid":"76561197974733729",
                    "communityvisibilitystate":3,
                    "profilestate":1,
                    "personaname":"slickity",
                    "lastlogoff":1548251594,
                    "commentpermission":2,
                    "profileurl":"https://steamcommunity.com/id/slickity/",
                    "avatar":"https://steamcdn-a.akamaihd.net/steamcommunity/public/images/avatars/6d/6d605500a24eb4f7016568662ba9ccd68fdbd514.jpg",
                    "avatarmedium":"https://steamcdn-a.akamaihd.net/steamcommunity/public/images/avatars/6d/6d605500a24eb4f7016568662ba9ccd68fdbd514_medium.jpg",
                    "avatarfull":"https://steamcdn-a.akamaihd.net/steamcommunity/public/images/avatars/6d/6d605500a24eb4f7016568662ba9ccd68fdbd514_full.jpg",
                    "personastate":3,
                    "primaryclanid":"103582791429523079",
                    "timecreated":1110343909,
                    "personastateflags":0
                }
            ]
        }
}
```

GetPlayerAchievements (v0001)

Returns a list of achievements for this user by app id. (Profile has to be public)

Example URL: http://api.steampowered.com/ISteamUserStats/GetPlayerAchievements/v0001/?appid=581320&key=389EC943738900A510BF540217AFB042&steamid=76561197972495328

Arguments

- steamid (64 bit Steam ID to return friend list for.)
- appid (The ID for the game you're requesting)
- **l** (Optional) (Language. If specified, it will return language data for the requested language.)

Sample (JSON): ???

GetUserStatsForGame (v0002)

Returns a list of achievements for this user by app id

Example URL: http://api.steampowered.com/ISteamUserStats/GetUserStatsForGame/v0002/?appid=440&key=389EC943738900A510BF540217AFB042&steamid=76561197972495328

Arguments

- steamid (64 bit Steam ID to return friend list for.)
- appid (The ID for the game you're requesting)
- **l** (Optional, Language. If specified, it will return language data for the requested language.)

Sample (JSON): **RETURNS ERROR 500 FOR NOW :/**

GetOwnedGames (v0001)

GetOwnedGames returns a list of games a player owns along with some playtime information, if the profile is publicly visible. Private, friends-only, and other privacy settings are not supported unless you are asking for your own personal details (ie the WebAPI key you are using is linked to the steamid you are requesting).

Example URL: http://api.steampowered.com/IPlayerService/GetOwnedGames/v0001/?key=389EC943738900A510BF540217AFB042&steamid=76561197974733729&format=json

Arguments

- steamid (The SteamID of the account.)
- include_appinfo (Include game name and logo information in the output. The default is to return appids only.)
- include_played_free_games (By default, free games like Team Fortress 2 are excluded (as technically everyone owns them). If include_played_free_games is set, they will be returned if the player has played them at some point. This is the same behavior as the games list on the Steam Community.)
- format (Output format. json (default), xml or vdf.)
- appids_filter (You can optionally filter the list to a set of appids. Note that these cannot be passed as a URL parameter, instead you must use the JSON format described in Steam_Web_API#Calling_Service_interfaces. The expected input is an array of integers (in JSON: "appids_filter: [440, 500, 550]"))

Sample (JSON):

```
{
    "response":{
        "game_count":134,
        "games":[
            {
                "appid":220,
                "playtime_forever":0
            },
            {
                "appid":240,
                "playtime_forever":0
            },
                    {
                "appid":843440,
                "playtime_forever":0
            },
            {
                "appid":860440,
                "playtime_forever":0
            },
            {
                "appid":880940,
                "playtime_forever":0
            },
            {
                "appid":350280,
                "playtime_forever":0
            },
            {
                "appid":884660,
                "playtime_forever":0
            },
            {
                "appid":581320,
                "playtime_forever":0
            }
        ]
    }
}
```

GetRecentlyPlayedGames (v0001)

GetRecentlyPlayedGames returns a list of games a player has played in the last two weeks, if the profile is publicly visible. Private, friends-only, and other privacy settings are not supported unless you are asking for your own personal details (ie the WebAPI key you are using is linked to the steamid you are requesting).

Example URL: http://api.steampowered.com/IPlayerService/GetRecentlyPlayedGames/v0001/?key=389EC943738900A510BF540217AFB042&steamid=76561197960434622&format=json

Arguments

- steamid (The SteamID of the account.)
- count (Optionally limit to a certain number of games (the number of games a person has played in the last 2 weeks is typically very small))
- format (Output format. json (default), xml or vdf.)

Sample (JSON):

```
{
    "response":{
        "total_count":2,
        "games":[
            {
                "appid":427520,
                "name":"Factorio",
                "playtime_2weeks":849,
                "playtime_forever":13377,
                "img_icon_url":"267f5a89f36ab287e600a4e7d4e73d3d11f0fd7d",
                "img_logo_url":"95e5aa627ae1481b1dea293f6db5954e8aa79f41"
            },
            {
                "appid":703980,
                "name":"Star Wars: Imperial Assault - Legends of the Alliance",
                "playtime_2weeks":5,
                "playtime_forever":286,
                "img_icon_url":"d2b66b97c2f7b3481fac60f30a9704bce5e21105",
                "img_logo_url":"b804759f75115d287b67c51ee368abd65c2cef4f"
            }
        ]
    }
}
```

GetPlayerBans (v1)

GetPlayerBans returns Community, VAC, and Economy ban statuses for given players.
Example URL: http://api.steampowered.com/ISteamUser/GetPlayerBans/v1/?key=389EC943738900A510BF540217AFB042&steamids=76561197974733729

Arguments

- steamids (Comma-delimited list of SteamIDs)

Result layout

- **players** List of player ban objects for each 64 bit ID requested
    - **SteamId (string)** The player's 64 bit ID.
    - **CommunityBanned (bool)** Indicates whether or not the player is banned from Steam Community.
    - **VACBanned (bool)** Indicates whether or not the player has VAC bans on record.
    - **NumberOfVACBans (int)** Number of VAC bans on record.
    - **DaysSinceLastBan (int)** Number of days since the last ban.
    - **NumberOfGameBans (int)** Number of bans in games, this includes CS:GO Overwatch bans.
    - **EconomyBan (string)** The player's ban status in the economy. If the player has no bans on record the string will be "none", if the player is on probation it will say "probation", etc.

Sample (JSON):

```
{
    "players":[
        {
            "SteamId":"76561197974733729",
            "CommunityBanned":false,
            "VACBanned":false,
            "NumberOfVACBans":0,
            "DaysSinceLastBan":0,
            "NumberOfGameBans":0,
            "EconomyBan":"none"
        }
    ]
}
```

GetNewsForApp (v0002)

GetNewsForApp returns the latest of a game specified by its appID.
Example URL: http://api.steampowered.com/ISteamNews/GetNewsForApp/v0002/?appid=440&count=3&maxlength=300&format=json

Arguments

- appid (AppID of the game you want the news of.)
- count (How many news enties you want to get returned.)
- maxlength (Maximum length of each news entry.)
- format (Output format. json (default), xml or vdf.)

Sample (JSON):

```
{
    "appnews":{
        "appid":440,
        "newsitems":[
            {
                "gid":"2426778849034624139",
                "title":"Team Fortress 2 Update Released",
                "url":"https://steamstore-a.akamaihd.net/news/externalpost/tf2_blog/2426778849034624139",
                "is_external_url":true,
                "author":"",
                "contents":"An update to Team Fortress 2 has been released. The update will be applied automatically
when you restart Team Fortress 2. The major changes include: * Improved memory usage on OS X and Linux systems;
* This should reduce the occurrence of \"Out of memory or address space\" errors on high texture qual...",
                "feedlabel":"TF2 Blog",
                "date":1548178020,
                "feedname":"tf2_blog",
                "feed_type":0,
                "appid":440
            },
            {
                "gid":"2426778849034623044",
                "title":"Team Fortress 2 Update Released",
                "url":"https://store.steampowered.com/news/47741/",
                "is_external_url":false,
                "author":"Valve",
                "contents":"An update to Team Fortress 2 has been released. The update will be applied automatically
when you restart Team Fortress 2. The major changes include: * Improved memory usage on OS X and Linux systems;
* This should reduce the occurrence of \"Out of memory or address space\" errors on high texture qual...",
                "feedlabel":"Product Update",
                "date":1548178020,
                "feedname":"steam_updates",
                "feed_type":0,
                "appid":440
            },
            {
                "gid":"2711631078218974515",
                "title":"Operation Madness vs Machines!",
                "url":"https://steamstore-a.akamaihd.net/news/externalpost/tf2_blog/2711631078218974515",
                "is_external_url":true,
                "author":"",
                "contents":"<a href=\"https://steamcommunity.com/groups/potatomvmservers#announcements/detail/
3539167513687424341\"> </a> <i>(Image credit - <a href=\"https://steamcommunity.com/profiles/
76561198045651005\">Mayor Rev</a>)</i> <a href=\"https://steamcommunity.com/groups/
potatomvmservers#announcements/detail/3539167513687424341\">Operation Madness vs Machines</a> is a brand
new community co-op Mann vs Machine charity event brought to you by the same group that created <a
href=\"https://steamcommunity.com/groups/potatomvmservers#announcements/detail/
1663388904536296299\">Operation Titanium Tank</a> and <a href=\"https://steamcommunity.com/groups/
potatomvmservers#announcements/detail/2533733726690659777\">Operation Canteen Crasher</a>! Operation
Madness vs Machines will take place on May 17th, 2019. Work closely w...",
                "feedlabel":"TF2 Blog",
                "date":1547252160,
                "feedname":"tf2_blog",
                "feed_type":0,
                "appid":440
            }
        ],
        "count":2600
    }
}
```

GetGlobalAchievementPercentagesForApp (v0002)

Returns on global achievements overview of a specific game in percentages.

Example: http://api.steampowered.com/ISteamUserStats/GetGlobalAchievementPercentagesForApp/v0002/?gameid=581320&format=json

Arguments

- gameid (AppID of the game you want the percentages of.)
- format (Output format. json (default), xml or vdf.)

```
{
    "achievementpercentages":{
        "achievements":[
            {
                "name":"ACH_WALLBANGER",
                "percent":75.9000015258789063
            },
            {
                "name":"ACH_POINT_SHOOTING",
                "percent":75.0999984741210938
            },
            {
                "name":"ACH_EXPLOITATIVE",
                "percent":66.4000015258789063
            },
            {
                "name":"ACH_TURF_WAR",
                "percent":56.2999992370605469
            },
            {
                "name":"ACH_HSLD",
                "percent":55.0999984741210938
            },
        ]
    }
}
```