

Forum:[Hack Forums](#)

Thread: [Twitch Username Availability Checker](#)

Threebones - January 8, 2017, 8:53 am

Inactive twitch names go available january 29th so I decided to make this. Reads names to check from a file, writes names that work to a file, works fast, and has proper error handling.

Get it here: <https://github.com/threebarber/twitch>

Threebones - January 8, 2017, 9:01 am

(Jan 8, 2017 09:54 AM)Jcb da Multi Wrote: [\[CITE\]](#) lol why'd u want a OG twitch... cool man

redstone - January 8, 2017, 11:30 am

Nice man! I had one on my old hard drive that got fucked up :(. I would take a look at some names but I am kind of bitter after Twitch banned my two letter account without reason and refused to do anything about it :(.

Xogg - January 8, 2017, 12:03 pm

Gonna use this as a inspiration, to go code my own, im going to make it in vb.net instead though

Aveling - January 8, 2017, 2:27 pm

Thank you for open sourcing this compact nifty script.

luke - January 8, 2017, 2:36 pm

Quite nice mate, will try and grab myself a name or two on the 29th ;)

Goodies - January 9, 2017, 1:07 am

You really over-complicated it, and in the process made it less readable and more slow.

- 1) The HTTP status code 404 is returned. You could have just checked the status code since they could feasibly change their 404 page at any time, but are not going to break HTTP standards.
- 2) It's synchronous and very slow. It is trivial to speed this up with simple multithreading.
- 3)

If you really want to use requests, then I definitely suggest using threading. The multiprocessing module has a little-known ([and not even document](#)) and yet incredibly useful thread pool class that can be used even before 2.7.

This code will run a lot faster and is, in my opinion, fairly readable. It can also run on Python 2.7/3.X.

PHP Code

```
<?php
from multiprocessing.pool import ThreadPool
import re
import requests
import sys
```

try:

```

input = raw_input
except NameError:
    pass

TWITCH_URL = "https://www.twitch.tv/{username}"
TWITCH_REGEX = re.compile(r"^[a-zA-Z0-9_]{4,25}$")
MAX_THREADS = 25
MESSAGES = {True: "Available", False: "Taken"}

def read_valid_usernames(filename):
    """Reads a list of usernames and filters out invalid ones."""
    try:
        with open(filename, "r") as fin:
            return [username for username in map(str.strip, fin) if TWITCH_REGEX.match(username)]
    except IOError:
        sys.exit("[!] '{}' - Invalid File".format(filename))

def username_available(username):
    """Checks if a 404 response code is given when requesting the profile. If it is, it is presumed to be available"""
    try:
        return username, requests.get(TWITCH_URL.format(username=username)).status_code == 404
    except Exception as e:
        print(e)

usernames = read_valid_usernames(input("Enter path to list of usernames: "))

for username, available in ThreadPool(MAX_THREADS).imap_unordered(username_available, usernames):
    print("{: <{size}}{}".format(username, MESSAGES.get(available, "Unknown"), size=len(max(usernames, key=len)) + 1))

```

Threebones - January 11, 2017, 11:21 am

(Jan 9, 2017 02:07 AM)Goodies Wrote: [\[CITE\]](#) You really over-complicated it, and in the process made it less readable and more slow.

- 1) The HTTP status code 404 is returned. You could have just checked the status code since they could feasibly change their 404 page at any time, but are not going to break HTTP standards.
- 2) It's synchronous and very slow. It is trivial to speed this up with simple multithreading.
- 3)

If you really want to use requests, then I definitely suggest using threading. The multiprocessing module has a little-known ([and not even document](#)) and yet incredibly useful thread pool class that can be used even before 2.7.

This code will run a lot faster and is, in my opinion, fairly readable. It can also run on Python 2.7/3.X.

PHP Code
<?php

```

from multiprocessing.pool import ThreadPool
import re
import requests
import sys

try:
    input = raw_input
except NameError:
    pass

TWITCH_URL = "https://www.twitch.tv/{username}"
TWITCH_REGEX = re.compile(r"^[a-zA-Z0-9_]{4,25}$")
MAX_THREADS = 25
MESSAGES = {True: "Available", False: "Taken"}

def read_valid_usernames(filename):
    """Reads a list of usernames and filters out invalid ones."""
    try:
        with open(filename, "r") as fin:
            return [username for username in map(str.strip, fin) if TWITCH_REGEX.match(username)]
    except IOError:
        sys.exit("[!] '{}' - Invalid File".format(filename))

def username_available(username):
    """Checks if a 404 response code is given when requesting the profile. If it is, it is presumed to be available"""
    try:
        return username, requests.get(TWITCH_URL.format(username=username)).status_code == 404
    except Exception as e:
        print(e)

usernames = read_valid_usernames(input("Enter path to list of usernames: "))

for username, available in ThreadPool(MAX_THREADS).imap_unordered(username_available, usernames):
    print("{:<{size}}{}".format(username, MESSAGES.get(available, "Unknown"), size=len(max(usernames, key=len)) + 1))

```

works more than fast enough for me, and I don't see how this is considered simplifying my code when what you posted looks 10x more complicated for something so trivial.

Goodies - January 11, 2017, 3:26 pm

(Jan 11, 2017 12:21 PM)Threebones Wrote: [\[CITE\]](#) works more than fast enough for me, and I don't see how this is considered simplifying my code when what you posted looks 10x more complicated for something so trivial.

There's really nothing that complex in there. And mine runs significantly faster than yours even can. It can request about ~25 at once... it runs roughly around 20x faster than yours. It can be increased with more threads.

Undefined - January 14, 2017, 3:13 am

(Jan 11, 2017 12:21 PM)Threebones Wrote: [\[CITE\]](#) works more than fast enough for me, and I don't

see how this is considered simplifying my code when what you posted looks 10x more complicated for something so trivial.

The one Goodies posted is actually very readable and not complex at all.

@Goodies, I've tested the checker with usernames that contain spaces in them ("this is an username" or "hello there") and it simply doesn't check them. I assume that is because Twitch disallows spaced usernames, right?

Goodies - January 14, 2017, 12:54 pm

(Jan 14, 2017 04:13 AM)Undefined Wrote: [\[CITE\]](#) @Goodies, I've tested the checker with usernames that contain spaces in them ("this is an username" or "hello there") and it simply doesn't check them. I assume that is because Twitch disallows spaced usernames, right?

I honestly have no idea. The only thing I'm checking is for a 404 status code. I don't even have a twitch account, and nor do I watch any streams. OP's code checks for a 404 page, so I changed it to looking at the status code. Other than that, it does more or less the same thing.

To be honest, though, I wouldn't even use the multithreaded version. I'd use aiohttp/asyncio. You'd limit the number of clients with a bounded semaphore and send the requests asynchronously. You'd get a fair amount of increased speed.

Undefined - January 14, 2017, 1:22 pm

(Jan 14, 2017 01:54 PM)Goodies Wrote: [\[CITE\]](#) I honestly have no idea. The only thing I'm checking is for a 404 status code. I don't even have a twitch account, and nor do I watch any streams. OP's code checks for a 404 page, so I changed it to looking at the status code. Other than that, it does more or less the same thing.

To be honest, though, I wouldn't even use the multithreaded version. I'd use aiohttp/asyncio. You'd limit the number of clients with a bounded semaphore and send the requests asynchronously. You'd get a fair amount of increased speed.

Yep, there's a problem with it actually. Since Twitch doesn't accept spaces anywhere in the username (even the end) pasted usernames might have issues. The script just doesn't even check them because Twitch declines them.

Check Discord - I've added something to the script. It now outputs the available usernames into a text file in the same location as the script/username list.

aVert - January 29, 2017, 9:57 am

thanks bro I will take a look at this

nooelec - January 30, 2017, 5:40 am

Using `.strip()` should remove trailing whitespace. If its spaces in the middle of the string, perhaps do `.replace(" ", "something else?")` to cut out/replace those spaces.

You know, you could probably apply the regex to brute force "short" Twitch usernames (i.e. 4 chars, which is the minimum, right?)...

Laski - March 9, 2017, 5:54 pm

You guys are doing it wrong, you're not supposed to check for a username after the "http://twitch.tv/" because it returns false positives. They have their own URL for checking usernames if you take a look at the headers. I've been using my checker for a bit and not one has shown as available. The method of checking for status codes is correct though.

Spoiler [\(Click to View\)](#)

This is in C# but the same applies for every language.

Invictum - *March 9, 2017, 6:04 pm*

Gonna try out this, thanks for sharing!