

Script to get the HTTP status code of a list of urls?

Asked 9 years, 9 months ago Active 10 months ago Viewed 133k times

I have a list of URLs that I need to check, to see if they still work or not. I would like to write a bash script that does that for me.

89 I only need the returned HTTP status code, i.e. 200, 404, 500 and so forth. Nothing more.

EDIT Note that there is an issue if the page says "404 not found" but returns a 200 OK message. It's a misconfigured web server, but you may have to consider this case.



47 For more on this, see [Check if a URL goes to a page containing the text "404"](#)



bash curl http-status-codes

Share Improve this question Follow

edited Nov 18 '18 at 7:14



codeforester
27.7k 9 74 104

asked May 26 '11 at 8:58



Manu
4,048 4 36 73

2 To be fair, my script's "bug" is only when the server returns HTTP code 200 but the body text says "404 not found", which is a misbehaving webserver. – [PhilR](#) Aug 4 '12 at 17:53

2 The exit status of wget will be 0 if the response code was 200, 8 if 404, 4 if 302... You can use the \$? variable to access the exit status of the previous command. – [Casey Watson](#) Dec 18 '13 at 22:14

8 Answers

Active

Oldest

Votes

Curl has a specific option, `--write-out`, for this:

199

```
$ curl -o /dev/null --silent --head --write-out '%{http_code}\n' <url>
200
```



- `-o /dev/null` throws away the usual output
- `--silent` throws away the progress meter
- `--head` makes a HEAD HTTP request, instead of GET
- `--write-out '%{http_code}\n'` prints the required status code

To wrap this up in a complete Bash script:

```
#!/bin/bash
while read LINE; do
    curl -o /dev/null --silent --head --write-out "%{http_code} $LINE\n" "$LINE"
done < url-list.txt
```

(Eagle-eyed readers will notice that this uses one curl process per URL, which imposes fork and TCP connection penalties. It would be faster if multiple URLs were combined in a single curl, but there isn't space to write out the monstrous repetition of options that curl requires to do this.)

Share Improve this answer Follow

edited Mar 25 '18 at 23:44

answered May 26 '11 at 10:07



PhilR
4,866 1 16 21

Very nice. Can I execute that command on every url in my file ? – [Manu](#) May 26 '11 at 10:40

1 @Manu: Yes, I've edited my answer to show one possible way of wrapping up the curl command. It assumes url-list.txt contains one URL per line. – [PhilR](#) May 26 '11 at 10:49

1 I don't know why script from above andswer always get me 000 in the output, but when I run command only once without loop it works... – [Karol F](#) Aug 9 '16 at 7:08

1 @KarolFiturski I had the same problem (which you've probably since fixed but just in case anyone else stumbles across this...) in my case I had carriage returns at the line ends of my input file, causing the urls to be like `http://example.com/\r` when going through the loop – [Jordan Robinson](#) Feb 21 '17 at 11:16

1 I had this issue and I was able to fix it by switching the line ending from the Windows type to the Linux type. – [Tristan](#) Jul 13 '17 at 1:40



40



prints only the status code for you



Share Improve this answer Follow

answered Feb 25 '12 at 10:40



user551168
459 5 2

9 +1 Shows multiple codes when a url is redirected, each at new line. – [Ashish Kumar - Ashfame](#) Apr 24 '12 at 21:55

Had to get rid of the `--spider` for it to work with the request that I was trying to make, but works. – [amitavk](#) Aug 6 '15 at 15:10

```
xargs -n1 -P 10 curl -o /dev/null --silent --head --write-out '%
{url_effective}': %{http_code}\n' < url.lst
```

-n1: use just one value (from the list) as argument to the curl call

-P10: Keep 10 curl processes alive at any time (i.e. 10 parallel connections)

Check the `write_out` parameter in the manual of curl for more data you can extract using it (times, etc).

In case it helps someone this is the call I'm currently using:

```
xargs -n1 -P 10 curl -o /dev/null --silent --head --write-out '%
{url_effective};%{http_code};%{time_total};%{time_namelookup};%{time_connect};%
{size_download};%{speed_download}\n' < url.lst | tee results.csv
```

It just outputs a bunch of data into a csv file that can be imported into any office tool.

Share Improve this answer Follow

answered Mar 13 '14 at 13:20



[estani](#)
16.9k 2 69 51

2 Parallelism, file input and csv. Exactly what i was looking for. – [Agey](#) Mar 2 '16 at 13:45

Brilliant, made my day. – [xltj](#) May 3 '17 at 6:39

This is awesome, just what I was looking for, thank you sir. One question, how could one include the page title of the page in the csv results? – [MitchellK](#) Jul 24 '17 at 12:47

@estani - stackoverflow.com/users/1182464/estani how could one include getting the page title of a page into the .csv file. Sorry for repost, forgot to tag you so you would get notified about this question. Many thanks. – [MitchellK](#) Jul 24 '17 at 13:32

@MitchellK this is not handling the contents of the http call at all. If the "page title" (whatever that is) is in the url, then you could add it. If not, you need to parse the whole page to extract the "title" of it (assuming you mean a html page retrieved by the http). Look for other answers at stack overflow or ask that specific question. – [estani](#) Jul 26 '17 at 13:02

This relies on widely available `wget`, present almost everywhere, even on Alpine Linux.

18

```
wget --server-response --spider --quiet "${url}" 2>&1 | awk 'NR==1{print $2}'
```

The explanations are as follow :

--quiet

Turn off Wget's output.

Source - [wget man pages](#)

--spider

[...] it will not download the pages, just check that they are there. [...]

Source - [wget man pages](#)

--server-response

Print the headers sent by HTTP servers and responses sent by FTP servers.

Source - [wget man pages](#)

What they don't say about `--server-response` is that those headers output are printed to [standard error \(stderr\)](#), thus the need to [redirect](#) to `stdin`.

The output sent to standard input, we can pipe it to `awk` to extract the HTTP status code. That code is :

- the second (`$2`) non-blank group of characters: `{ $2 }`
- on the very first line of the header: `NR==1`

And because we want to print it... `{print $2}` .

```
wget --server-response --spider --quiet "${url}" 2>&1 | awk 'NR==1{print $2}'
```

Share Improve this answer Follow

answered Nov 18 '18 at 5:25



[Salathiel Genèse](#)
1,406 1 17 25

1 I used this one with `2>&1 | head -1 | awk '{ print $2 }'` – [Evhz](#) Sep 20 '19 at 12:14

Use [curl](#) to fetch the HTTP-header only (not the whole file) and parse it:



Share Improve this answer Follow

answered May 26 '11 at 9:25



[dogbane](#)
239k 71 367 393

curl tells me 200 when wget says 404 ... (– [Manu](#) Jun 22 '11 at 15:19)

The `-I` flag causes curl to make a HTTP HEAD request, which is treated separately from a normal HTTP GET by some servers and can thus return different values. The command should still work without it. – [lambshaanxy](#) Dec 28 '12 at 10:52



wget -S -i *file* will get you the headers from each url in a file.

4

Filter though `grep` for the status code specifically.



Share Improve this answer Follow



answered May 26 '11 at 9:10



[colinross](#)
2,005 12 10



I found a tool "webchk" written in Python. Returns a status code for a list of urls. <https://pypi.org/project/webchk/>

1

Output looks like this:



```
▶ webchk -i ./dxieu.txt | grep '200'
http://salesforce-case-status.dxi.eu/login ... 200 OK (0.108)
https://support.dxi.eu/hc/en-gb ... 200 OK (0.389)
https://support.dxi.eu/hc/en-gb ... 200 OK (0.401)
```



Hope that helps!

Share Improve this answer Follow

answered Apr 29 '20 at 21:05



[Yura Loginov](#)
680 7 6



Due to https://mywiki.woledge.org/BashPitfalls#Non-atomic_writes_with_xargs_-P (output from parallel jobs in `xargs` risks being mixed), I would use GNU Parallel instead of `xargs` to parallelize:

0



```
cat url.lst |
parallel -P0 -q curl -o /dev/null --silent --head --write-out '%
{url_effective}: %{http_code}\n' > outfile
```



In this particular case it may be safe to use `xargs` because the output is so short, so the problem with using `xargs` is rather that if someone later changes the code to do something bigger, it will no longer be safe. Or if someone reads this question and thinks he can replace `curl` with something else, then that may also not be safe.

Share Improve this answer Follow

answered Sep 7 '19 at 6:36



[Ole Tange](#)
26.8k 5 70 85



Highly active question. Earn 10 reputation in order to answer this question. The reputation requirement helps protect this question from spam and non-answer activity.