

Forum: [Hack Forums](#)

Thread: [Check if a site exists within a second](#)

Skidy - April 12, 2018, 10:33 am

Checking if a website exists it easy, but if it doesn't exist it takes ages to verify.

E.g

```
Code from urllib.request import Request, urlopen
from urllib.error import URLError, HTTPError
req = Request("http://stackoverflow.com")
try:
    response = urlopen(req)
except HTTPError as e:
    print('The server couldn\'t fulfill the request.')
    print('Error code: ', e.code)
except URLError as e:
    print('We failed to reach a server.')
    print('Reason: ', e.reason)
else:
    print('Website is working fine')
```

If the site doesn't exist, it takes ages to verify it. I need to it check within 2 seconds. Any ideas?

Fluxify - April 12, 2018, 11:09 am

I just tried your script, and when entering a random website I receive this response within a second:

```
We failed to reach a server.
Reason: [Errno 11001] getaddrinfo failed
```

Which reason are you getting when using a random non-existing website?

.kr. - April 12, 2018, 12:33 pm

```
use urlopen('url', timeout=2)
```

i've tested this with a long polling request. it give's a timeout. website exist

```
Code from urllib.request import Request, urlopen
from urllib.error import URLError, HTTPError
from socket import timeout
```

```
def main():
    req = Request("http://domain.net/test/sleep.php")
    try:
        response = urlopen(req, timeout=2)
    except HTTPError as e:
        print('The server couldn\'t fulfill the request.')
        print('Error code: ', e.code)
    except URLError as e:
```

```

    print('We failed to reach a server.')
    print('Reason: ', e.reason)
except timeout as e:
    print('Socket timeout.')
else:
    print('Website is working fine')

```

```

if __name__ == '__main__':
    main()

```

MichalMonday - April 13, 2018, 12:57 am

you could use head request instead (e.g. by using requests library)

Quote: The HEAD method is identical to GET except that the server MUST NOT return a message-body in the response. The meta-information contained in the HTTP headers in response to a HEAD request SHOULD be identical to the information sent in response to a GET request. This method can be used for obtaining meta-information about the entity implied by the request without transferring the entity-body itself. This method is often used for testing hypertext links for validity, accessibility, and recent modification.

YooGotti - April 13, 2018, 12:59 am

(Apr 12, 2018 12:33 PM).kr. Wrote: [CITE](#) use urlopen('url', timeout=2)

i've tested this with a long polling request. it give's a timeout. website exist

```

Code from urllib.request import Request, urlopen
from urllib.error import URLError, HTTPError
from socket import timeout

```

```

def main():
    req = Request("")
    try:
        response = urlopen(req, timeout=2)
    except HTTPError as e:
        print('The server couldn\'t fulfill the request.')
        print('Error code: ', e.code)
    except URLError as e:
        print('We failed to reach a server.')
        print('Reason: ', e.reason)
    except timeout as e:
        print('Socket timeout.')
    else:
        print('Website is working fine')

```

```

if __name__ == '__main__':
    main()

```

same i did the same, idk