

How would I create a Python web server that downloads a file on any GET request?

Asked 8 years, 1 month ago Active 1 year, 1 month ago Viewed 11k times



4



1



I am attempting to create an easy solution to file sharing across computers in my local network. I used to execute `python -m SimpleHTTPServer` in bash whenever I wanted to share a directory, but I wanted a way to share only one specific file. Can anyone point me in the right direction for how I might create a web server, and then have it download a file for each GET request. For example, someone on my network could go to my IP and have the file download.

P.S. What would be even cooler is if there was a way to password protect a file! Also, I have Python 2.7.2, if that matters. Anyway, As you have probably noticed, I know almost nothing about Python, but I learn by example so I am hoping this will help me some as well.

Thanks a bunch in advance!

`python` `file` `http` `web` `webserver`

Share Improve this question Follow

asked Aug 31 '13 at 1:38



russellsayshi

2,111 ● 5 ● 15 ● 20

COLLECTIVES™
on stackoverflow

Join communities
for your favorite
technologies

GitLab Google Cloud GO

[Report this ad](#)

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Customize settings



```
try:
    import http.server as BaseHTTPServer # Python 3.x
except ImportError:
    import BaseHTTPServer # Python 2.x
import os
import shutil
import sys

FILEPATH = sys.argv[1] if sys.argv[1:] else __file__

class SimpleHTTPRequestHandler(BaseHTTPServer.BaseHTTPRequestHandler):
    def do_GET(self):
        with open(FILEPATH, 'rb') as f:
            self.send_response(200)
            self.send_header("Content-Type", 'application/octet-stream')
            self.send_header("Content-Disposition", 'attachment; filename="
{}"'.format(os.path.basename(FILEPATH)))
            fs = os.fstat(f.fileno())
            self.send_header("Content-Length", str(fs.st_size))
            self.end_headers()
            shutil.copyfileobj(f, self.wfile)

def test(handler_class=SimpleHTTPRequestHandler,
         server_class=BaseHTTPServer.HTTPServer,
         protocol="HTTP/1.0"):
    if sys.argv[2:]:
        port = int(sys.argv[2])
    else:
        port = 8000
    server_address = ('', port)

    handler_class.protocol_version = protocol
    httpd = BaseHTTPServer.HTTPServer(server_address, handler_class)

    sa = httpd.socket.getsockname()
    print("Serving HTTP on {}[0] port {}[1] ... {}".format(sa, FILEPATH))
    httpd.serve_forever()

if __name__ == '__main__':
    test()
```

Usage: `python script_path [filepath_to_serve [port]]`

UPDATE code works for Python 2.x / 3.x

Share Improve this answer Follow

edited Feb 6 '18 at 15:38

answered Aug 31 '13 at 1:50



[falsetru](#)

324k ● 51 ● 634 ● 569

1 @russellsayshi I don't mind, I referred to `SimpleHTTPServer.py` source code - [falsetru](#) Aug 31 '13 at

Your privacy

By clicking "Accept all cookies", you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Customize settings

an | findstr :8000 if you use Windows). It should print line similar to tcp 0 0

0.0.0.0:8000

0.0.0.0:*

LISTEN . – [falsetru](#) Aug 31 '13 at 2:22 

1 [@SasukeUchiha](#), I updated the code to work both on Python 2/3. – [falsetru](#) Feb 6 '18 at 15:38

Your privacy

By clicking “Accept all cookies”, you agree Stack Exchange can store cookies on your device and disclose information in accordance with our [Cookie Policy](#).

Accept all cookies

Customize settings