

Computer Network Laboratory

Programming with HTTP

Jiawei Chang

Dept. of Computer Science and Information Engineering
National Taichung University of Science and Technology

Outline

1. `download_webpage`
2. `simple_http_server`
3. `http_compression`
4. `proxy_web_request`
5. `checking_webpage_with_HEAD_request`
6. `spoof_mozilla_firefox_in_client_code`
7. `http_fail_over_client`

DOWNLOAD_WEBPAGE

Download Webpage

```
# This program requires Python 3.5.2 or any later version
# It may run on any other version with/without modifications.
#
# Follow the comments inline to make it run on Python 2.7.x.

import argparse

import urllib
# Comment out the above line and uncomment the below for Python 2.7.x.
#import urllib2

REMOTE_SERVER_HOST = 'http://www.cnn.com'

class HTTPClient:

    def __init__(self, host):
        self.host = host

    def fetch(self):
        response = urllib.urlopen(self.host)
        # Comment out the above line and uncomment the below for Python 2.7.x.
        #response = urllib2.urlopen(self.host)

        data = response.read()
        text = data.decode('utf-8')
        return text

if __name__ == "__main__":
    parser = argparse.ArgumentParser(description='HTTP Client Example')
    parser.add_argument('--host', action="store", dest="host", default=REMOTE_SERVER_HOST)

    given_args = parser.parse_args()
    host = given_args.host
    client = HTTPClient(host)
    print (client.fetch())
```

Download Webpage

```

File Edit View Search Terminal Help
jwchang@ubuntu: ~/Desktop/20190425

="m-footer_list-item"><a class="m-footer_link" data-analytics="footer_vr_archives" href="/specials/vr/vr-archives">Archives/</li></ol></li></ol><ol class="m-footer_bucket m-footer_bucket_intl more">
<li class="m-footer_title"><a class="m-footer_title_link" href="/more" data-analytics="footer_intl_more">More/</a><ol class="m-footer_subtitles m-footer_subtitles_intl more">
<li class="m-footer_title"><a class="m-footer_title_link" data-analytics="footer_intl_more_weather" href="/weather">Weather/</li><li class="m-footer_title"><a class="m-footer_title_link" data-analytics="footer_intl_more_cnn_profiles" href="/specials/profiles">CNN Profiles/</a></li><li class="m-footer_title"><a class="m-footer_title_link" data-analytics="footer_intl_more_leadership" href="/specials/more/cnn-leadership">Leadership/</li><li class="m-footer_title"><a class="m-footer_title_link" data-analytics="footer_intl_more_pressroom" href="/cnnpressroom.blogs.cnn.com">Pressroom/</li><li class="m-footer_title"><a class="m-footer_title_link" data-analytics="footer_intl_more_partner_sites" href="/specials/world/specials-page-cnn-partners">Partner sites/</li><li class="m-footer_title"><a class="m-footer_title_link" data-analytics="footer_intl_more_work_for_cnn" href="https://www.turnerjobs.com/search-jobs?orgIds=1174&ac=19299">Work for CNN/</li></ol></li></ol><ol class="m-footer_bucket m-footer_bucket_widget">
<li></li></ol><div class="m-copyright-outercontainer"><div class="edition-picker"><a class="logo" href="/" data-analytics="edition-picker-logo"></a><div class="drop-down"><div class="current">International Edition/</div><div class="list js-edition-picker_list-set" data-location="footer" data-analytics="edition-picker"><li class="item">Set edition preference/</li><li class="item">option " data-values="https://us.cnn.com">input id="edition-picker-edition-footer" type="radio" class="js-edition-preferences_choice" name="edition-pref-footer" value="www"><label for="edition-picker-edition-footer">U.S. </label></li><li class="item">option selected" data-value="https://edition.cnn.com">input id="edition-picker-www-footer" type="radio" class="js-edition-preferences_choice" name="edition-pref-www" value="edition"><label for="edition-picker-www-footer">International/</li><li class="item">confirm js-edition-preferences_confirm">button class="edition-picker_confirm-button">Confirm/</li><li class="item">edition" data-type="www" data-value="https://us.cnn.com">U.S. </li><li class="item">edition" data-type="edition" data-value="https://edition.cnn.com">International/</li><li class="item">edition" data-value="https://arabic.cnn.com">Arabic/</li><li class="item">edition" data-value="http://cnnespanol.cnn.com">Espaol/</li></ol></div><div class="m-copyright-innercontainer"><div class="m-copyright"><span class="copyright">&copy; &nbsp;1989&nbsp;1998&nbsp;2018&nbsp;2019&nbsp;2020&nbsp;2021&nbsp;2022&nbsp;2023&nbsp;2024&nbsp;2025&nbsp;2026&nbsp;2027&nbsp;2028&nbsp;2029&nbsp;2030&nbsp;2031&nbsp;2032&nbsp;2033&nbsp;2034&nbsp;2035&nbsp;2036&nbsp;2037&nbsp;2038&nbsp;2039&nbsp;2040&nbsp;2041&nbsp;2042&nbsp;2043&nbsp;2044&nbsp;2045&nbsp;2046&nbsp;2047&nbsp;2048&nbsp;2049&nbsp;2050&nbsp;2051&nbsp;2052&nbsp;2053&nbsp;2054&nbsp;2055&nbsp;2056&nbsp;2057&nbsp;2058&nbsp;2059&nbsp;2060&nbsp;2061&nbsp;2062&nbsp;2063&nbsp;2064&nbsp;2065&nbsp;2066&nbsp;2067&nbsp;2068&nbsp;2069&nbsp;2070&nbsp;2071&nbsp;2072&nbsp;2073&nbsp;2074&nbsp;2075&nbsp;2076&nbsp;2077&nbsp;2078&nbsp;2079&nbsp;2080&nbsp;2081&nbsp;2082&nbsp;2083&nbsp;2084&nbsp;2085&nbsp;2086&nbsp;2087&nbsp;2088&nbsp;2089&nbsp;2090&nbsp;2091&nbsp;2092&nbsp;2093&nbsp;2094&nbsp;2095&nbsp;2096&nbsp;2097&nbsp;2098&nbsp;2099&nbsp;2100&nbsp;2101&nbsp;2102&nbsp;2103&nbsp;2104&nbsp;2105&nbsp;2106&nbsp;2107&nbsp;2108&nbsp;2109&nbsp;2110&nbsp;2111&nbsp;2112&nbsp;2113&nbsp;2114&nbsp;2115&nbsp;2116&nbsp;2117&nbsp;2118&nbsp;2119&nbsp;2120&nbsp;2121&nbsp;2122&nbsp;2123&nbsp;2124&nbsp;2125&nbsp;2126&nbsp;2127&nbsp;2128&nbsp;2129&nbsp;2130&nbsp;2131&nbsp;2132&nbsp;2133&nbsp;2134&nbsp;2135&nbsp;2136&nbsp;2137&nbsp;2138&nbsp;2139&nbsp;2140&nbsp;2141&nbsp;2142&nbsp;2143&nbsp;2144&nbsp;2145&nbsp;2146&nbsp;2147&nbsp;2148&nbsp;2149&nbsp;2150&nbsp;2151&nbsp;2152&nbsp;2153&nbsp;2154&nbsp;2155&nbsp;2156&nbsp;2157&nbsp;2158&nbsp;2159&nbsp;2160&nbsp;2161&nbsp;2162&nbsp;2163&nbsp;2164&nbsp;2165&nbsp;2166&nbsp;2167&nbsp;2168&nbsp;2169&nbsp;2170&nbsp;2171&nbsp;2172&nbsp;2173&nbsp;2174&nbsp;2175&nbsp;2176&nbsp;2177&nbsp;2178&nbsp;2179&nbsp;2180&nbsp;2181&nbsp;2182&nbsp;2183&nbsp;2184&nbsp;2185&nbsp;2186&nbsp;2187&nbsp;2188&nbsp;2189&nbsp;2190&nbsp;2191&nbsp;2192&nbsp;2193&nbsp;2194&nbsp;2195&nbsp;2196&nbsp;2197&nbsp;2198&nbsp;2199&nbsp;2200&nbsp;2201&nbsp;2202&nbsp;2203&nbsp;2204&nbsp;2205&nbsp;2206&nbsp;2207&nbsp;2208&nbsp;2209&nbsp;2210&nbsp;2211&nbsp;2212&nbsp;2213&nbsp;2214&nbsp;2215&nbsp;2216&nbsp;2217&nbsp;2218&nbsp;2219&nbsp;2220&nbsp;2221&nbsp;2222&nbsp;2223&nbsp;2224&nbsp;2225&nbsp;2226&nbsp;2227&nbsp;2228&nbsp;2229&nbsp;2230&nbsp;2231&nbsp;2232&nbsp;2233&nbsp;2234&nbsp;2235&nbsp;2236&nbsp;2237&nbsp;2238&nbsp;2239&nbsp;2240&nbsp;2241&nbsp;2242&nbsp;2243&nbsp;2244&nbsp;2245&nbsp;2246&nbsp;2247&nbsp;2248&nbsp;2249&nbsp;2250&nbsp;2251&nbsp;2252&nbsp;2253&nbsp;2254&nbsp;2255&nbsp;2256&nbsp;2257&nbsp;2258&nbsp;2259&nbsp;2260&nbsp;2261&nbsp;2262&nbsp;2263&nbsp;2264&nbsp;2265&nbsp;2266&nbsp;2267&nbsp;2268&nbsp;2269&nbsp;2270&nbsp;2271&nbsp;2272&nbsp;2273&nbsp;2274&nbsp;2275&nbsp;2276&nbsp;2277&nbsp;2278&nbsp;2279&nbsp;2280&nbsp;2281&nbsp;2282&nbsp;2283&nbsp;2284&nbsp;2285&nbsp;2286&nbsp;2287&nbsp;2288&nbsp;2289&nbsp;2290&nbsp;2291&nbsp;2292&nbsp;2293&nbsp;2294&nbsp;2295&nbsp;2296&nbsp;2297&nbsp;2298&nbsp;2299&nbsp;2300&nbsp;2301&nbsp;2302&nbsp;2303&nbsp;2304&nbsp;2305&nbsp;2306&nbsp;2307&nbsp;2308&nbsp;2309&nbsp;2310&nbsp;2311&nbsp;2312&nbsp;2313&nbsp;2314&nbsp;2315&nbsp;2316&nbsp;2317&nbsp;2318&nbsp;2319&nbsp;2320&nbsp;2321&nbsp;2322&nbsp;2323&nbsp;2324&nbsp;2325&nbsp;2326&nbsp;2327&nbsp;2328&nbsp;2329&nbsp;2330&nbsp;2331&nbsp;2332&nbsp;2333&nbsp;2334&nbsp;2335&nbsp;2336&nbsp;2337&nbsp;2338&nbsp;2339&nbsp;2340&nbsp;2341&nbsp;2342&nbsp;2343&nbsp;2344&nbsp;2345&nbsp;2346&nbsp;2347&nbsp;2348&nbsp;2349&nbsp;2350&nbsp;2351&nbsp;2352&nbsp;2353&nbsp;2354&nbsp;2355&nbsp;2356&nbsp;2357&nbsp;2358&nbsp;2359&nbsp;2360&nbsp;2361&nbsp;2362&nbsp;2363&nbsp;2364&nbsp;2365&nbsp;2366&nbsp;2367&nbsp;2368&nbsp;2369&nbsp;2370&nbsp;2371&nbsp;2372&nbsp;2373&nbsp;2374&nbsp;2375&nbsp;2376&nbsp;2377&nbsp;2378&nbsp;2379&nbsp;2380&nbsp;2381&nbsp;2382&nbsp;2383&nbsp;2384&nbsp;2385&nbsp;2386&nbsp;2387&nbsp;2388&nbsp;2389&nbsp;2390&nbsp;2391&nbsp;2392&nbsp;2393&nbsp;2394&nbsp;2395&nbsp;2396&nbsp;239
```

SIMPLE_HTTP_SERVER

simple_http_server

```
#!/usr/bin/env python
# This program requires Python 3.5.2 or any later version
# It may run on any other version with/without modifications.
#
# Follow the comments inline to make it run on Python 2.7.x.

import argparse
import sys

from http.server import BaseHTTPRequestHandler, HTTPServer
# Comment out the above line and uncomment the below for Python 2.7.x.
#from BaseHTTPServer import BaseHTTPRequestHandler, HTTPServer

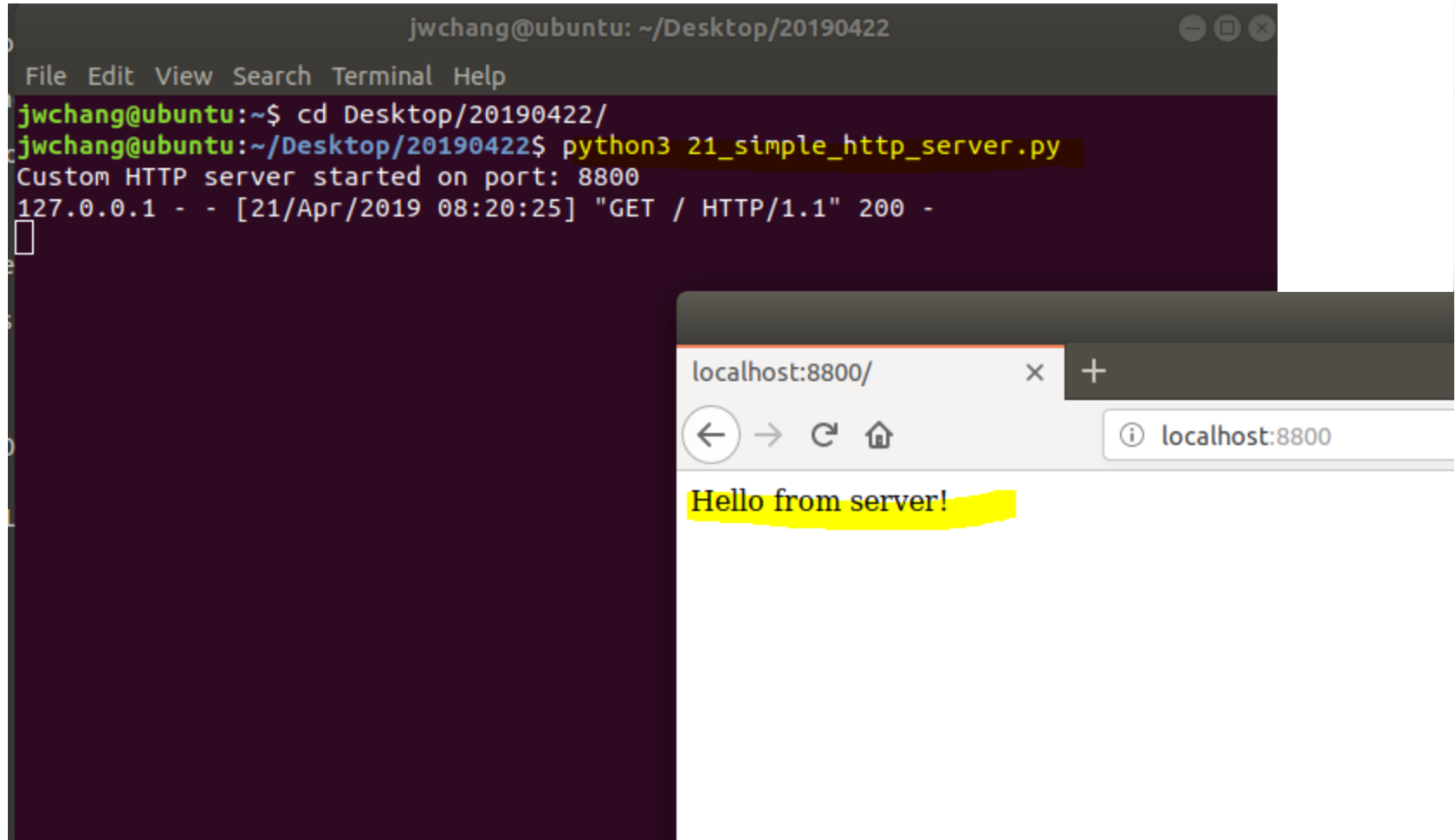
DEFAULT_HOST = '127.0.0.1'
DEFAULT_PORT = 8800

class RequestHandler(BaseHTTPRequestHandler):
    """ Custom request handler """

    def do_GET(self):
        """ Handler for the GET requests """
        self.send_response(200)
        self.send_header('Content-type', 'text/html')
        self.end_headers()
        # Send the message to browser
        self.wfile.write("Hello from server!".encode())
        return

class CustomHTTPServer(HTTPServer):
    "A custom HTTP server"
    def __init__(self, host, port):
        server_address = (host, port)
        HTTPServer.__init__(self, server_address, RequestHandler)
```

simple_http_server



The image shows a terminal window and a web browser. The terminal window, titled 'jwchang@ubuntu: ~/Desktop/20190422', displays the following commands and output:

```
jwchang@ubuntu:~$ cd Desktop/20190422/  
jwchang@ubuntu:~/Desktop/20190422$ python3 21_simple_http_server.py  
Custom HTTP server started on port: 8800  
127.0.0.1 - - [21/Apr/2019 08:20:25] "GET / HTTP/1.1" 200 -
```

The web browser window shows the address bar with 'localhost:8800/' and the page content 'Hello from server!'.

HTTP_COMPRESSION

http_compression

```
import argparse
import string
import os
import sys
import gzip

import io
# Comment out the above line and uncomment the below for Python 2.7.x.
#import cStringIO

from http.server import BaseHTTPRequestHandler, HTTPServer
# Comment out the above line and uncomment the below for Python 2.7.x.
#from BaseHTTPServer import BaseHTTPRequestHandler, HTTPServer

DEFAULT_HOST = '127.0.0.1'
DEFAULT_PORT = 8800

HTML_CONTENT = b"""<html><body><h1>Compressed Hello World!</h1></body></html>"""
# Comment out the above line and uncomment the below for Python 2.7.x.
#HTML_CONTENT = b"""<html><body><h1>Compressed Hello World!</h1></body></html>"""
```

http_compression

```
class RequestHandler(BaseHTTPRequestHandler):
    """ Custom request handler """

    def do_GET(self):
        """ Handler for the GET requests """
        self.send_response(200)
        self.send_header('Content-type', 'text/html')
        self.send_header('Content-Encoding', 'gzip') |
        zbuf = self.compress_buffer(HTML_CONTENT)
        sys.stdout.write("Content-Encoding: gzip\r\n")
        self.send_header('Content-Length', len(zbuf))
        self.end_headers()
        # Send the message to browser
        zbuf = self.compress_buffer(HTML_CONTENT)
        sys.stdout.write("Content-Encoding: gzip\r\n")
        sys.stdout.write("Content-Length: %d\r\n" % (len(zbuf)))
        sys.stdout.write("\r\n")

        self.wfile.write(zbuf)

    return

def compress_buffer(self, buf):

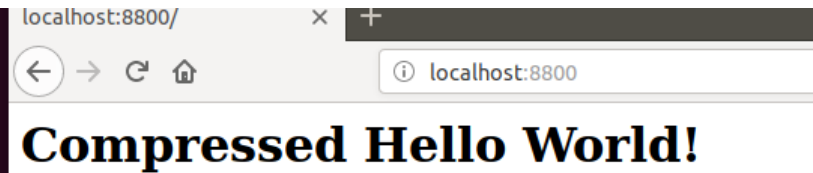
    zbuf = io.BytesIO()
    # Comment out the above line and uncomment the below for Python 2.7.x.
    #zbuf = cStringIO.StringIO()

    zfile = gzip.GzipFile(mode = 'wb', fileobj = zbuf, compresslevel = 6)
    zfile.write(buf)
    zfile.close()
    return zbuf.getvalue()

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='Simple HTTP Server Example')
    parser.add_argument('--port', action="store", dest="port", type=int, default=DEFAULT_PORT)
    given_args = parser.parse_args()
    port = given_args.port
    server_address = (DEFAULT_HOST, port)
    server = HTTPServer(server_address, RequestHandler)
    server.serve_forever()
```

http_compression

```
jwchang@ubuntu:~/Desktop/20190422$ python3 25_http_compression.py  
127.0.0.1 - - [21/Apr/2019 08:42:12] "GET / HTTP/1.1" 200 -  
Content-Encoding: gzip  
Content-Encoding: gzip  
Content-Length: 71
```



PROXY_WEB_REQUEST

proxy_web_request

```
#!/usr/bin/env python
# This program requires Python 3.5.2 or any later version
# It may run on any other version with/without modifications.
#
# Follow the comments inline to make it run on Python 2.7.x.

import urllib.request, urllib.parse, urllib.error
# Comment out the above line and uncomment the below for Python 2.7.x.
#import urllib

URL = 'https://www.github.com'
PROXY_ADDRESS = "165.24.10.8:8080" # By Googling free proxy server

if __name__ == '__main__':

    proxy = urllib.request.ProxyHandler({"http" : PROXY_ADDRESS})
    opener = urllib.request.build_opener(proxy)
    urllib.request.install_opener(opener)
    resp = urllib.request.urlopen(URL)
    # Comment out the above 4 lines and uncomment the below for Python 2.7.x.
    #resp = urllib.urlopen(URL, proxies = {"http" : PROXY_ADDRESS})

    print ("Proxy server returns response headers: %s " %resp.headers)
```

proxy_web_request

```
Proxy server returns response headers: Server: GitHub.com
Date: Sun, 21 Apr 2019 15:27:13 GMT
Content-Type: text/html; charset=utf-8
Transfer-Encoding: chunked
Connection: close
Status: 200 OK
Vary: X-PJAX
ETag: W/"e5fc906228cab0d3ba6a54a476dface1"
Cache-Control: max-age=0, private, must-revalidate
Set-Cookie: has_recent_activity=1; path=/; expires=Sun, 21 Apr 2019 16:27:13 -0000
Set-Cookie: _octo=GH1.1.715440119.1555860433; domain=.github.com; path=/; expires=Wed, 21 Apr 2021 15:27:13 -0000
Set-Cookie: logged_in=no; domain=.github.com; path=/; expires=Thu, 21 Apr 2039 15:27:13 -0000; secure; HttpOnly
Set-Cookie: _gh_sess=UFFRTDhqZ2JGRmR2Q1RkTnRLc0x5ZEw3SytoeG1hdjVIZS9IWLpLL3pqd25VdHR0YTB4Wld3eVJwOVY1VERWQjJvZDJPY1BNbXNsRwDL3MkVic2NtNTh3M3V2eVlPNnNRT2doSVBxbEc1WGp6UnJIZTB6YWxDaUtCN2lqdUFzNnMwN1FlQ1NzU3JHeFRSSctjRFYyTWNCrjQ3dz09LS13TkrUvkdCNVpkSHBMttpOnly
X-Request-Id: 84375b3f-0873-4299-b454-3773a74b3c37
Strict-Transport-Security: max-age=31536000; includeSubdomains; preload
X-Frame-Options: deny
X-Content-Type-Options: nosniff
X-XSS-Protection: 1; mode=block
Referrer-Policy: origin-when-cross-origin, strict-origin-when-cross-origin
Expect-CT: max-age=2592000, report-uri="https://api.github.com/_private/browser/errors"
Content-Security-Policy: default-src 'none'; base-uri 'self'; block-all-mixed-content; connect-src 'self' uploads.github.com s.com github-cloud.s3.amazonaws.com github-production-repository-file-5c1aeb.s3.amazonaws.com github-production-upload-manifest-com wss://live.github.com; font-src github.githubassets.com; form-action 'self' github.com gist.github.com; frame-ancestors thubassets.com identities.github.com collector.githubapp.com github-cloud.s3.amazonaws.com *.githubusercontent.com customer-sb.githubassets.com; style-src 'unsafe-inline' github.githubassets.com
X-GitHub-Request-Id: 22DF:0468:4F7060:835243:5CBC8BD0
```

CHECKING_WEBPAGE_WITH _HEAD_REQUEST

checking_webpage_with_HEAD_request

```
import argparse

import http.client
# Comment out the above line and uncomment the below for Python 2.7.x.
#import httplib

import urllib.parse
# Comment out the above line and uncomment the below for Python 2.7.x.
#import urlparse

import re
import urllib.request, urllib.error
# Comment out the above line and uncomment the below for Python 2.7.x.
#import urllib

DEFAULT_URL = 'http://www.python.org'

HTTP_GOOD_CODES = [http.client.OK, http.client.FOUND, http.client.MOVED_PERMANENTLY]
# Comment out the above line and uncomment the below for Python 2.7.x.
#HTTP_GOOD_CODES = [httplib.OK, httplib.FOUND, httplib.MOVED_PERMANENTLY]

def get_server_status_code(url):
    """
    Download just the header of a URL and
    return the server's status code.
    """
    host, path = urllib.parse.urlparse(url)[1:3]
    # Comment out the above line and uncomment the below for Python 2.7.x.
    #host, path = urlparse.urlparse(url)[1:3]
    try:
        conn = http.client.HTTPConnection(host)
        # Comment out the above line and uncomment the below for Python 2.7.x.
        #conn = httplib.HTTPConnection(host)

        conn.request('HEAD', path)
        return conn.getresponse().status

    except Exception as e:
        print ("Server: %s status is: %s " %(url, e))
        # Comment out the above line and uncomment the below for Python 2.7.x.
        #except StandardError:
        return None

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='Example HEAD Request')
    parser.add_argument('--url', action="store", dest="url", default=DEFAULT_URL)
    given_args = parser.parse_args()
    url = given_args.url
    if get_server_status_code(url) in HTTP_GOOD_CODES:
        print ("Server: %s status is OK: %s " %(url, get_server_status_code(url)))
    else:
        print ("Server: %s status is NOT OK: %s" %(url, get_server_status_code(url)))
```

checking_webpage_with_HEAD_ request

```
Server: http://www.python.org status is OK: 301
```

**SPOOF_MOZILLA_FIREFOX_IN_
CLIENT_CODE**

spoofer_mozilla_firefox_in_client_code

```
#!/usr/bin/env python
# This program requires Python 3.5.2 or any later version
# It may run on any other version with/without modifications.
#
# Follow the comments inline to make it run on Python 2.7.x.

import urllib.request, urllib.error, urllib.parse
# Comment out the above line and uncomment the below for Python 2.7.x.
#import urllib2

BROWSER = 'Mozilla/5.0 (Windows NT 5.1; rv:20.0) Gecko/20100101 Firefox/20.0'
URL = 'http://www.python.org'

def spoof_firefox():

    opener = urllib.request.build_opener()
    # Comment out the above line and uncomment the below for Python 2.7.x.
    #opener = urllib2.build_opener()

    opener.addheaders = [('User-agent', BROWSER)]
    result = opener.open(URL)
    print ("Response headers:")

    for header in result.headers:
        # Comment out the above line and uncomment the below for Python 2.7.x.
        #for header in result.headers.headers:
            print ("%s: %s" %(header, result.headers.get(header)))
            # Comment out the above line and uncomment the below for Python 2.7.x.
            #print (header)

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

spoofer_mozilla_firefox_in_client_code

```
Response headers:
Server: nginx
Content-Type: text/html; charset=utf-8
X-Frame-Options: DENY
Via: 1.1 vegur
Via: 1.1 vegur
Content-Length: 48182
Accept-Ranges: bytes
Date: Sun, 21 Apr 2019 15:35:19 GMT
Via: 1.1 vegur
Age: 812
Connection: close
X-Served-By: cache-iad2124-IAD, cache-tyo19947-TYO
X-Cache: HIT, HIT
X-Cache-Hits: 2, 983
X-Timer: S1555860919.296719,VS0,VE0
Vary: Cookie
Strict-Transport-Security: max-age=63072000; includeSubDomains
```

HTTP_FAIL_OVER_CLIENT

http_fail_over_client

```
import urllib.request, urllib.parse, urllib.error
import os

TARGET_URL = 'http://python.org/ftp/python/2.7.4/'
TARGET_FILE = 'Python-2.7.4.tgz'

class CustomURLopener(urllib.request.FancyURLopener):
    # Comment out the above line and uncomment the below for Python 2.7.x.
    #class CustomURLopener(urllib.FancyURLopener):
        """Override FancyURLopener to skip error 206 (when a
        partial file is being sent)
        """
        def http_error_206(self, url, fp, errcode, errmsg, headers, data=None):
            pass
```

http_fail_over_client

```
def resume_download():
    file_exists = False
    CustomURLClass = CustomURL opener()
    if os.path.exists(TARGET_FILE):
        out_file = open(TARGET_FILE, "ab")
        file_exists = os.path.getsize(TARGET_FILE)
        #If the file exists, then only download the unfinished part
        CustomURLClass.addheader("range", "bytes=%s-" % (file_exists))
    else:
        out_file = open(TARGET_FILE, "wb")

    web_page = CustomURLClass.open(TARGET_URL + TARGET_FILE)

    #Check if last download was OK
    if int(web_page.headers['Content-Length']) == file_exists:
        loop = 0
        print ("File already downloaded!")

    byte_count = 0
    while True:
        data = web_page.read(8192)
        if not data:
            break
        out_file.write(data)
        byte_count = byte_count + len(data)

    web_page.close()
    out_file.close()

    for k,v in list(web_page.headers.items()):
        # Comment out the above line and uncomment the below for Python 2.7.x.
        #for k,v in web_page.headers.items():
            print (k, "=", v)
    print ("File copied", byte_count, "bytes from", web_page.url)

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


http_fail_over_client

```
Server = nginx
Content-Type = application/octet-stream
Last-Modified = Sat, 06 Apr 2013 14:16:10 GMT
ETag = "51602e2a-dd15e7"
X-Clacks-Overhead = GNU Terry Pratchett
Via = 1.1 varnish
Accept-Ranges = bytes
Age = 20734695
Content-Length = 14489063
Accept-Ranges = bytes
Date = Sun, 21 Apr 2019 15:45:53 GMT
Via = 1.1 varnish
Connection = close
X-Served-By = cache-iad2130-IAD, cache-hnd18738-HND
X-Cache = HIT, HIT
X-Cache-Hits = 0, 0
X-Timer = S1555861554.667674,VS0,VE2
Strict-Transport-Security = max-age=63072000; includeSubDomains
File copied 14489063 bytes from http://www.python.org/ftp/python/2.7.4/Python-2.7.4.tgz
```

延伸閱讀

- Socket Programming in Python (Guide)
 - <https://realpython.com/python-sockets/#socket-api-overview>
- Python 网络编程
 - <http://www.runoob.com/python/python-socket.html>

Resource is available by
<https://jiaweichang.github.io/biography/>

THANKS