COMS 4119 Computer Networks (Spring 2021)
Problem Set #2

Linxiao Wu - lw2944@columbia.edu

02/16/2021

Problem 7

I am using a MAC to do my following tests:

Make a GET request

First, I will use my terminal to sent a get request as follows

1. Run the proxy_server.py on port 8888

2. use terminal to test

```
curl -X GET "http://httpbin.org/get" -H "accept: application/json" --
                                      proxy 127.0.0.1:8888
```

3. As there is no cache in the file, the request in the proxy server prints out

```
Making a GET request.....
Create a socket on the proxyserver
remote host name: httpbin.org
Connect to the socket to port 80...
b'HTTP/1.1 200 OK\r\nDate: Sun, 28 Feb 2021 19:25:20 GMT\r\nContent-
                                      Type: application/json\r\
                                      nContent-Length: 306\r\
                                      nConnection: keep-alive\r\
                                      nServer: gunicorn/19.9.0\r\
                                      nAccess-Control-Allow-Origin: *\
                                      r\nAccess-Control-Allow-
                                      Credentials: true\r\n\r\n{\n  "
                                      args": {}, \n  "headers": {\n
                                         "Accept": "application/json",
                                       \n    "Host": "httpbin.org", \n
                                          "Proxy-Connection": "Keep-
                                      Alive", \n    "User-Agent": "
                                      curl/7.71.1", \n    "X-Amzn-
                                      Trace-Id": "Root=1-603bee20-
                                      25ba36587435649215 11c919"\n  },
                                      \n  "origin": "74.71.224.150", \
                                      n  "url": "http://httpbin.org/
                                      get"\n}\n'
```

4. Then I try go.com

```
curl -X GET "http://go.com" --proxy 127.0.0.1:8888
```

5. I get response

```
curl -X GET "http://go.com" --proxy 127.0.0.1:8888
<!DOCTYPE html>
<!--

    "We keep moving forward, opening up new doors and
      doing new things, because we're curious ...
        and curiosity keeps leading us down new paths."

                              Walt Disney

-->
<html class="no-js" version="HTML+RDFa 1.1" lang="en">
<head prefix="og: http://ogp.me/ns#" dir="ltr">
  <title>Go.com | The Walt Disney Company </title>
  <script src="https://static-mh.content.disney.io/matterhorn/assets/
                                        jquery-dc8b911876ec.js" type="
                                        text/javascript"></script>
  <link rel="preconnect" href="//static-mh.content.disney.io">
<link rel="preconnect" href="//lumiere-a.akamaihd.net">
<link rel="preconnect" href="//kaltura.akamaized.net">
<link rel="preconnect" href="//cdnapisec.kaltura.com">
<link rel="preconnect" href="//a.dilcdn.com">
<link rel="preconnect" href="//assets.adobedtm.com">
<link rel="preload" href="https://static-mh.content.disney.io/
                                      matterhorn/assets/matterhorn/
                                      sans/matterhorn-regular-
                                      eed88f0756d6.woff" as="font"
                                      type="font/woff" crossorigin="
                                      anonymous">
<link rel="preload" href="https://static-mh.content.disney.io/
                                      matterhorn/assets/application-
                                      bfac960f836d.css" as="style"
                                      type="text/css" crossorigin="
                                      anonymous">
<link rel="preload" href="https://static-mh.content.disney.io/
                                      matterhorn/assets/matterhorn-v3-
                                      player-skin-77381cd10564.css" as
                                      ="style" type="text/css"
                                      crossorigin="anonymous">
<link rel="preload" href="//global.go.com/stat/dolWebAnalytics.js" as
                                      ="script" type="application/
                                      javascript" crossorigin="
                                      anonymous">
```

```
<link rel="stylesheet" href="https://static-mh.content.disney.io/
                                     matterhorn/assets/application-
                                     bfac960f836d.css" type="text/css
                                     ">
<link rel="stylesheet" href="https://static-mh.content.disney.io/
                                     matterhorn/assets/modules/
                                     background_styles-1903f7131478.
                                     css" type="text/css">
<link rel="stylesheet" href="https://static-mh.content.disney.io/
                                     matterhorn/assets/modules/
                                     rich_image-7ac318d6c07a.css"
                                     type="text/css">
<link rel="stylesheet" href="https://static-mh.content.disney.io/
                                     matterhorn/assets/modules/
                                     rich_image-7ac318d6c07a.css"
                                     type="text/css">
<link rel="stylesheet" href="https://static-mh.content.disney.io/
                                     matterhorn/assets/modules/
                                     rich_text-fc52ad2df661.css" type
                                     ="text/css">
curl: (18) transfer closed with 597033 bytes remaining to read
<link rel="stylesheet" type="text/css" href="https://static-mh.
                                     content.disney.io/matterhorn/
                                     assets/goc/wide-011049455401.css
                                     "><link rel="stylesheet" type="
                                     text/css" href="https://static-
                                     mh.content.disney.io/matterhorn/
                                     assets/goc/wide-desktop-
                                     4e2ca4cc1f2a.css" media="all and
                                      (min%
```

6. Then I try google.com

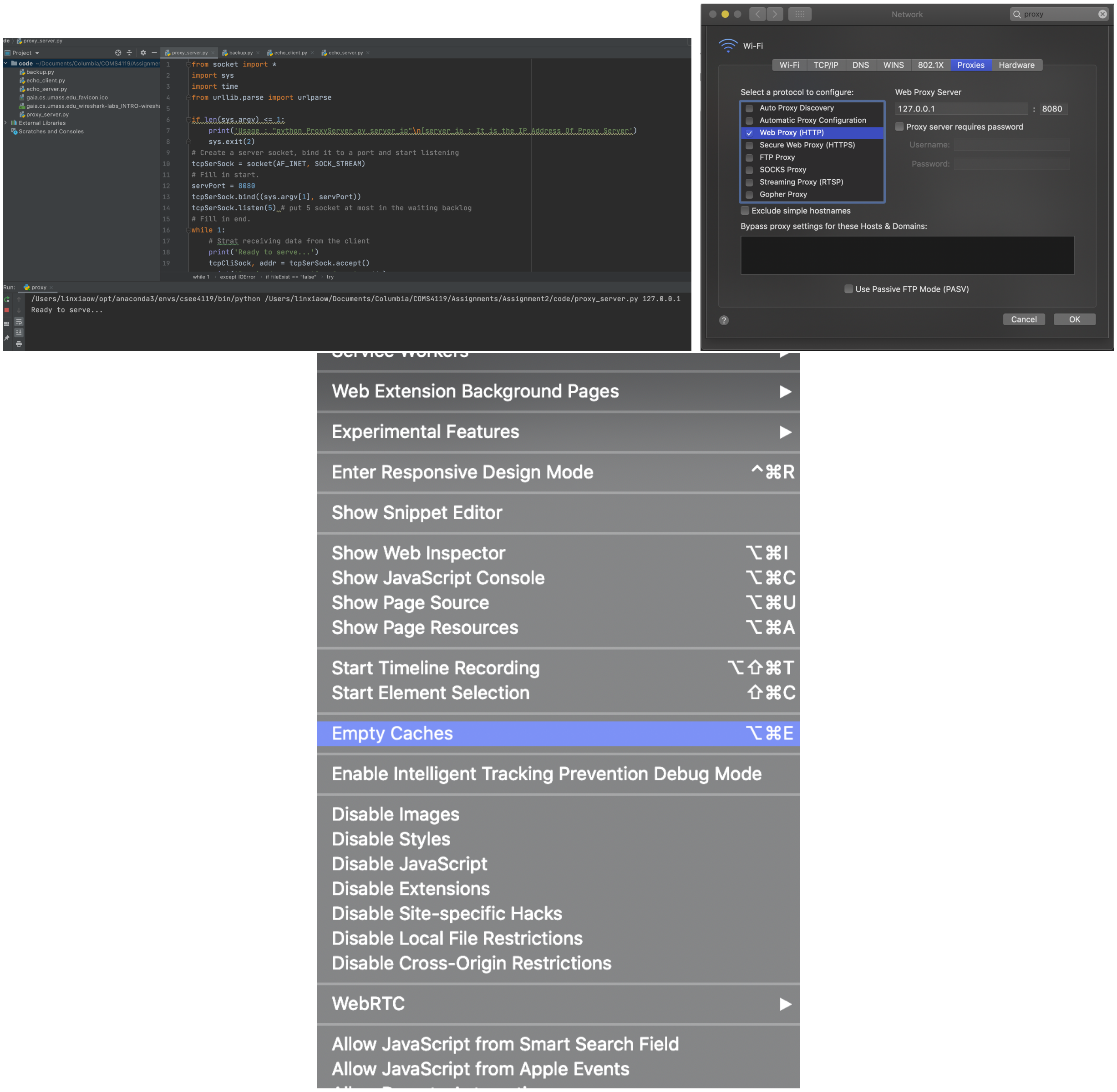
```
curl -X GET "http://google.com" --proxy 127.0.0.1:8888
```

7. I get response that it has been moved

```
<HTML><HEAD><meta http-equiv="content-type" content="text/html;
                                     charset=utf-8">
<TITLE>301 Moved</TITLE></HEAD><BODY>
<H1>301 Moved</H1>
The document has moved
<A HREF="http://www.google.com/">here</A>.
</BODY></HTML>
```

Proxy the Request in Web

I will show how I make a request via my proxy server.
Set up: Run the Proxy server, and set up the Web Proxy to the port I specified, and also clear the cache
Then, I type the url `http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html` to the Safari website and get the webpage as shown in Figure 2



```
from socket import *
import sys
import time
from urllib.parse import urlparse

if len(sys.argv) <= 1:
    print('Usage : "python ProxyServer.py server_ip"\n[server_ip : It is the IP Address Of Proxy Server')
    sys.exit(2)
# Create a server socket, bind it to a port and start listening
tcpSerSock = socket(AF_INET, SOCK_STREAM)
# Fill in start.
servPort = 8080
tcpSerSock.bind((sys.argv[1], servPort))
tcpSerSock.listen(5) # put 5 socket at most in the waiting backlog
# Fill in end.
while 1:
    # Strat receiving data from the client
    print('Ready to serve...')
    tcpCliSock, addr = tcpSerSock.accept()
```

```
/Users/linxiaow/opt/anaconda3/envs/csee4119/bin/python /Users/linxiaow/Documents/Columbia/COMS4119/Assignments/Assignment2/code/proxy_server.py 127.0.0.1
Ready to serve...
```

Select a protocol to configure:

- □ Auto Proxy Discovery
- □ Automatic Proxy Configuration
- ☑ Web Proxy (HTTP)
- □ Secure Web Proxy (HTTPS)
- □ FTP Proxy
- □ SOCKS Proxy
- □ Streaming Proxy (RTSP)
- □ Gopher Proxy

Web Proxy Server: 127.0.0.1 : 8080

□ Proxy server requires password

Username:

Password:

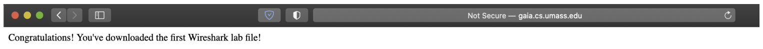
□ Exclude simple hostnames

Bypass proxy settings for these Hosts & Domains:

□ Use Passive FTP Mode (PASV)

proxy_server.py
Project
backup.py
echo_client.py
echo_server.py
code ~/Documents/Columbia/COMS4119/Assignmer
gaia.cs.umass.edu_favicon.ico
gaia.cs.umass.edu_wireshark-labs_INTRO-wiresha
External Libraries
Scratches and Consoles
while 1 › except IOError › if fileExist == "false" › try
Run: proxy
Network
proxy
Wi-Fi
Wi-Fi TCP/IP DNS WINS 802.1X Proxies Hardware
?
Cancel
OK
Web Extension Background Pages ▶
Experimental Features ▶
Enter Responsive Design Mode ^⌘R
Show Snippet Editor
Show Web Inspector ⌥⌘I
Show JavaScript Console ⌥⌘C
Show Page Source ⌥⌘U
Show Page Resources ⌥⌘A
Start Timeline Recording ⌥⇧⌘T
Start Element Selection ⇧⌘C
Empty Caches ⌥⌘E
Enable Intelligent Tracking Prevention Debug Mode
Disable Images
Disable Styles
Disable JavaScript
Disable Extensions
Disable Site-specific Hacks
Disable Local File Restrictions
Disable Cross-Origin Restrictions
WebRTC ▶
Allow JavaScript from Smart Search Field
Allow JavaScript from Apple Events

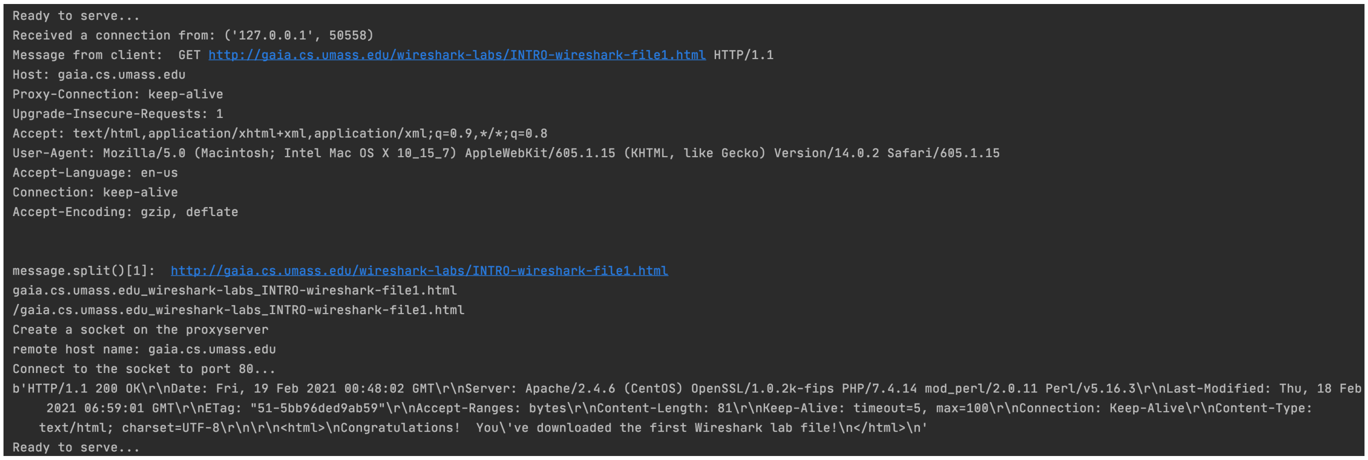
Figure 1: Configure the Proxy Server



Congratulations! You've downloaded the first Wireshark lab file!

Figure 2: Web Page Content

Also, the socket information shows as the Figure 3. As there is no cache, the Proxy server makes a request to host `gaia.cs.umass.edu` to get the content and then return to the client.



```
Ready to serve...
Received a connection from: ('127.0.0.1', 50558)
Message from client:  GET http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html HTTP/1.1
Host: gaia.cs.umass.edu
Proxy-Connection: keep-alive
Upgrade-Insecure-Requests: 1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.0.2 Safari/605.1.15
Accept-Language: en-us
Connection: keep-alive
Accept-Encoding: gzip, deflate


message.split()[1]:  http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html
gaia.cs.umass.edu_wireshark-labs_INTRO-wireshark-file1.html
/gaia.cs.umass.edu_wireshark-labs_INTRO-wireshark-file1.html
Create a socket on the proxyserver
remote host name: gaia.cs.umass.edu
Connect to the socket to port 80...
b'HTTP/1.1 200 OK\r\nDate: Fri, 19 Feb 2021 00:48:02 GMT\r\nServer: Apache/2.4.6 (CentOS) OpenSSL/1.0.2k-fips PHP/7.4.14 mod_perl/2.0.11 Perl/v5.16.3\r\nLast-Modified: Thu, 18 Feb
     2021 06:59:01 GMT\r\nETag: "51-5bb96ded9ab59"\r\nAccept-Ranges: bytes\r\nContent-Length: 81\r\nKeep-Alive: timeout=5, max=100\r\nConnection: Keep-Alive\r\nContent-Type:
    text/html; charset=UTF-8\r\n\r\n<html>\nCongratulations!  You\'ve downloaded the first Wireshark lab file!\n</html>\n'
Ready to serve...
```

Figure 3: First Get Request: There is no cache, so a remote socket client is created

Optional 1: 404

I assume when there is connection errors, it is 404

```
except gaierror as e:
    print("Hostname doesn’t have IP Address" + hostn)
    print("reason: ", str(e))
    tcpCliSock.send("HTTP/1.1 404 Not Found\r\n".encode())
    tcpCliSock.send("\r\n".encode())
    tcpCliSock.send("\r\n".encode())

except ConnectionRefusedError as e:
    print("Invalid Hostname" + hostn)
    print("reason: ", str(e))
    tcpCliSock.send("HTTP/1.1 404 Not Found\r\n".encode())
    tcpCliSock.send("\r\n".encode())
    tcpCliSock.send("\r\n".encode())
```

When I go to a web page which is reference in `http://roxanageambasu.github.io/ds-class//assets/lectures/lecture4.pdf` and is no longer available, it will be displayed as 404 in the Figure 4

Option 2: POST

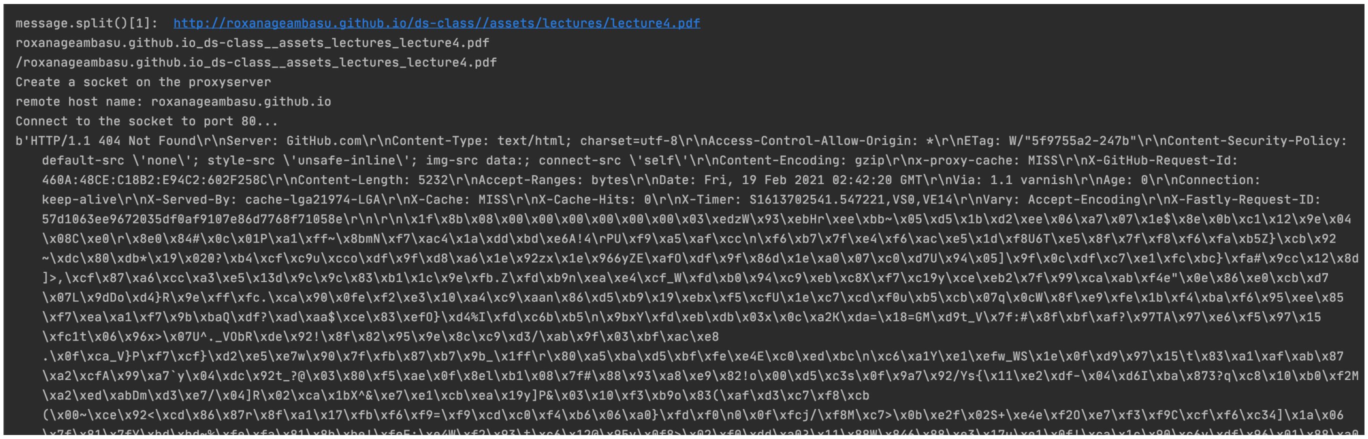
I will use my terminal for this functionality to test

1. I make a post to http.org

```
curl -X POST "http://httpbin.org/post" -H "accept: application/json"
                                               --proxy 127.0.0.1:8888
```

2. My proxy will automatically make the post request to the server

```
Making a POST request.....
Create a socket on the proxyserver
```



```
message.split()[1]:  http://roxanageambasu.github.io/ds-class//assets/lectures/lecture4.pdf
roxanageambasu.github.io_ds-class__assets_lectures_lecture4.pdf
/roxanageambasu.github.io_ds-class__assets_lectures_lecture4.pdf
Create a socket on the proxyserver
remote host name: roxanageambasu.github.io
Connect to the socket to port 80...
b'HTTP/1.1 404 Not Found\r\nServer: GitHub.com\r\nContent-Type: text/html; charset=utf-8\r\nAccess-Control-Allow-Origin: *\r\nETag: W/"5f9755a2-247b"\r\nContent-Security-Policy:
    default-src \'none\'; style-src \'unsafe-inline\'; img-src data:; connect-src \'self\'\r\nContent-Encoding: gzip\r\nx-proxy-cache: MISS\r\nX-GitHub-Request-Id:
    460A:48CE:C18B2:E94C2:602F258C\r\nContent-Length: 5232\r\nAccept-Ranges: bytes\r\nDate: Fri, 19 Feb 2021 02:42:20 GMT\r\nVia: 1.1 varnish\r\nAge: 0\r\nConnection:
    keep-alive\r\nX-Served-By: cache-lga21974-LGA\r\nX-Cache: MISS\r\nX-Cache-Hits: 0\r\nX-Timer: S1613702541.547221,VS0,VE14\r\nVary: Accept-Encoding\r\nX-Fastly-Request-ID:
    57d1063ee9672035df0af9107e86d7768f71058e\r\n\r\n\x1f\x8b\x08\x00\x00\x00\x00\x00\x00\x03\xedzW\x93\xebHr\xee\xbb~\x05\xd5\x1b\xd2\xee\x06\xa7\x07\x1e$\x8e\x0b\xc1\x12\x9e\x04
    \x08C\xe0\r\x8e0\x84#\x0c\x01P\xa1\xff~\x8bmN\xf7\xac4\x1a\xdd\xbd\xe6A!4\rPU\xf9\xa5\xaf\xcc\n\xf6\xb7\x7f\xe4\xf6\xac\xe5\x1d\xf8U6T\xe5\x8f\x7f\xf8\xf6\xfa\xb5Z}\xcb\x92
    ~\xdc\x80\xdb*\x19\x020?\xb4\xcf\xc9u\xcco\xdf\x9f\xd8\xa6\x1e\x92zx\x1e\x966yZE\xafO\xdf\x9f\x86d\x1e\xa0\x07\xc0\xd7U\x94\x05]\x9f\x0c\xdf\xc7\xe1\xfc\xbc}\xfa#\x9cc\x12\x8d
    ]>,\xcf\x87\xa6\xcc\xa3\xe5\x13d\x9c\x9c\x83\xb1\x1c\x9e\xfb.Z\xfd\xb9n\xea\xe4\xcf_W\xfd\xb0\x94\xc9\xeb\xc8X\xf7\xc19y\xce\xeb2\x7f\x99\xca\xab\xf4e"\x0e\x86\xe0\xcb\xd7
    \x07L\x9dDo\xd4}R\x9e\xff\xfc.\xca\x90\x0fe\xf2\xe3\x10\xa4\xc9\xaan\x86\xd5\xb9\x19\xebx\xf5\xcfU\x1e\xc7\xcd\xf0u\xb5\xcb\x07q\x0cW\x8f\xe9\xfe\x1b\xf4\xba\xf6\x95\xee\x85
    \xf7\xea\xa1\xf7\x9b\xbaQ\xdf?\xad\xaa$\xce\x83\xefO}\xd4%I\xfd\xc6b\xb5\n\x9bxY\xfd\xeb\xdb\x03x\x0c\xa2K\xda=\x18=GM\xd9t_V\x7f:#\x8f\xbf\xaf?\x97TA\x97\xe6\xf5\x97\x15
    \xfc1t\x06\x96x>\x07U^._VObR\xde\x92!\x8f\x82\x95\x9e\x8c\xc9\xd3/\xab\x9f\x03\xbf\xac\xe8
    .\x0f\xca_V}P\xf7\xcf}\xd2\xe5\xe7w\x90\x7f\xfb\x87\xb7\x9b_\x1ff\r\x80\xa5\xba\xd5\xbf\xfe\xe4E\xc0\xed\xbc\n\xc6\xa1Y\xe1\xefw_WS\x1e\x0f\xd9\x97\x15\t\x83\xa1\xaf\xab\x87
    \xa2\xcfA\x99\xa7`y\x04\xdc\x92t_?@\x03\x80\xf5\xae\x0f\x8el\xb1\x08\x7f#\x88\x93\xa8\xe9\x82!o\x00\xd5\xc3s\x0f\x9a7\x92/Ys{\x11\xe2\xdf-\x04\xd6I\xba\x873?q\xc8\x10\xb0\xf2M
    \xa2\xed\xabDm\xd3\xe7/\x04]R\x02\xca\x1bX^&\xe7\xe1\xcb\xea\x19y]P&\x03\x10\xf3\xb9o\x83(\xaf\xd3\xc7\xf8\xcb
    (\x00~\xce\x92<\xcd\x86\x87r\x8f\xa1\x17\xfb\xf6\xf9=\xf9\xcd\xc0\xf4\xb6\x06\xa0}\xfd\xf0\n0\x0f\xfcj/\xf8M\xc7>\x0b\xe2f\x02S+\xe4e\xf20\xe7\xf3\xf9C\xcf\xf6\xc34]\x1a\x06
```

Figure 4: Go to a webpage no longer available

```
remote host name: httpbin.org
Connect to the socket to port 80...
b'HTTP/1.1 200 OK\r\nDate: Sun, 28 Feb 2021 20:41:36 GMT\r\nContent-
                                      Type: application/json\r\
                                      nContent-Length: 370\r\
                                      nConnection: keep-alive\r\
                                      nServer: gunicorn/19.9.0\r\
                                      nAccess-Control-Allow-Origin: *\
                                      r\nAccess-Control-Allow-
                                      Credentials: true\r\n\r\n{\n  "
                                      args": {}, \n  "data": "", \n  "
                                      files": {}, \n  "form": {}, \n
                                      "headers": {\n    "Accept": "
                                      application/json", \n    "Host":
                                       "httpbin.org", \n    "Proxy-
                                      Connection": "Keep-Alive", \n
                                         "User-Agent": "curl/7.71.1",
                                      \n    "X-Amzn-Trace-Id": "Root=1
                                      -603c0000-
                                      3be8e9ea301584107a5f285b"\n  },
                                      \n  "json": null, \n  "origin":
                                      "74.71.224.150", \n  "url": "
                                      http://httpbin.org/post"\n}\n'
Ready to serve...
```

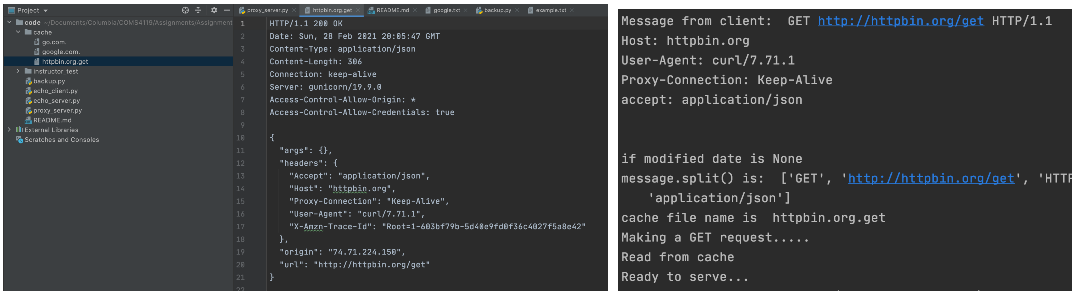
Option 3: Caching

As usual, I test in the terminal with the same request

1. make request to http://httpbin.org/get again

```
curl -X GET "http://httpbin.org/get" -H "accept: application/json" --
                                      proxy 127.0.0.1:8888
```

2. As in the previous section and there is no ”If-Modified-Since” header and this file is cached, so return the cache which is also indicated by the console



```
HTTP/1.1 200 OK
Date: Sun, 28 Feb 2021 20:05:47 GMT
Content-Type: application/json
Content-Length: 306
Connection: keep-alive
Server: gunicorn/19.9.0
Access-Control-Allow-Origin: *
Access-Control-Allow-Credentials: true

{
  "args": {},
  "headers": {
    "Accept": "application/json",
    "Host": "httpbin.org",
    "Proxy-Connection": "Keep-Alive",
    "User-Agent": "curl/7.71.1",
    "X-Amzn-Trace-Id": "Root=1-603bf79b-5d40e9fd0f36c4027f5a8e42"
  },
  "origin": "74.71.224.150",
  "url": "http://httpbin.org/get"
}
```

```
Message from client:  GET http://httpbin.org/get HTTP/1.1
Host: httpbin.org
User-Agent: curl/7.71.1
Proxy-Connection: Keep-Alive
accept: application/json


if modified date is None
message.split() is:  ['GET', 'http://httpbin.org/get', 'HTTP
    'application/json']
cache file name is  httpbin.org.get
Making a GET request.....
Read from cache
Ready to serve...
```

Figure 5: Cached content

Then I will present how I use If-Modified-Since header.
I will use the `go.com` to show my example
Here is what I have in proxy cache:

```
# cache for go.com
HTTP/1.1 200 OK
date: Sun, 28 Feb 2021 19:41:00 GMT
content-type: text/html;charset=utf-8
content-length: 599493
cache-control: public, max-age=1812
x-xss-protection: 1; mode=block
x-content-type-options: nosniff
x-ua-compatible: IE=edge,chrome=1
etag: W/"9f3384677cd7562724f5af346e2ae7e5"
age: 1798
accept-ranges: bytes
x-server: px-mh-ha-f
```

as we can see the date, which I assume is the update time in the proxy side

1. I send a request with the if modified date:

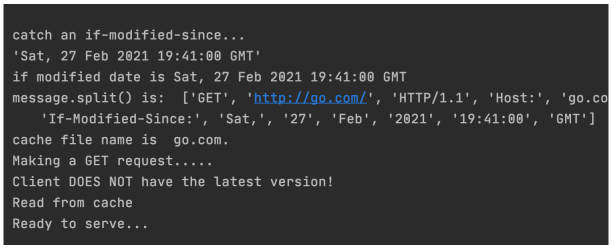
   ```
   curl -v GET "http://go.com" -H "If-Modified-Since: Sat, 27 Feb 2021
                                    19:41:00 GMT" --proxy 127.0.0.1:
                                    8888
   ```

2. As you can see, the client side does not have a latest version

3. Then I make another request with 304 shown in the below from my proxy server

   ```
   curl -v GET "http://go.com" -H "If-Modified-Since: Sun, 28 Feb 2021
                                    19:41:00 GMT" --proxy 127.0.0.1:
                                    8888
   ```

4. as this is up-to-date, the server responds with



```
catch an if-modified-since...
'Sat, 27 Feb 2021 19:41:00 GMT'
if modified date is Sat, 27 Feb 2021 19:41:00 GMT
message.split() is:  ['GET', 'http://go.com/', 'HTTP/1.1', 'Host:', 'go.co
   'If-Modified-Since:', 'Sat,', '27', 'Feb', '2021', '19:41:00', 'GMT']
cache file name is  go.com.
Making a GET request.....
Client DOES NOT have the latest version!
Read from cache
Ready to serve...
```

Figure 6: Not up to date, response with cache

```
    curl -v GET "http://go.com" -H "If-Modified-Since: Sun, 28 Feb
                                          2021 19:41:00 GMT" --proxy 127.0
                                          .0.1:8888

*   Trying 127.0.0.1:8888...
* Connected to 127.0.0.1 (127.0.0.1) port 8888 (#0)
> GET http://GET/ HTTP/1.1
> Host: GET
> User-Agent: curl/7.71.1
> Accept: */*
> Proxy-Connection: Keep-Alive
> If-Modified-Since: Sun, 28 Feb 2021 19:41:00 GMT
>
* Empty reply from server
* Connection #0 to host 127.0.0.1 left intact
curl: (52) Empty reply from server
* Found bundle for host 127.0.0.1: 0x7ff43941a640 [serially]
* Re-using existing connection! (#0) with proxy 127.0.0.1
* Connected to 127.0.0.1 (127.0.0.1) port 8888 (#0)
> GET http://go.com/ HTTP/1.1
> Host: go.com
> User-Agent: curl/7.71.1
> Accept: */*
> Proxy-Connection: Keep-Alive
> If-Modified-Since: Sun, 28 Feb 2021 19:41:00 GMT
>
* Recv failure: Connection reset by peer
* Connection died, retrying a fresh connect
* Closing connection 0
* Issue another request to this URL: 'http://go.com/'
* Hostname 127.0.0.1 was found in DNS cache
*   Trying 127.0.0.1:8888...
* Connected to 127.0.0.1 (127.0.0.1) port 8888 (#1)
> GET http://go.com/ HTTP/1.1
> Host: go.com
> User-Agent: curl/7.71.1
> Accept: */*
> Proxy-Connection: Keep-Alive
> If-Modified-Since: Sun, 28 Feb 2021 19:41:00 GMT
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 304 Not Modified
```

```
* Connection #1 to host 127.0.0.1 left intact
```