DNS_HttpProxy_assignment

A DNS and HTTP Proxy server from scratch, using low level socket programming in Python.

A multi-threaded web proxy.

This web proxy can only handle GET request for now.

File Structure

```
.
├── Cache
|    ├── [Some_cached_website_elements_files]
├── my_proxyServer.py
```

Usage: python3 my_proxyServer.py [server_ip_address] [server_listening_port_number]

Example:

```
python3 my_proxyServer.py 0.0.0.0 8080

# To test it, simply open a browser to access http://localhost:8080/www.google.com or http://[server_ip_address]:8080/www.google.com
```

A DNS server resolving www.yourlastname.432.(In this case, www.xie.432.)

File structure

```
.
├── ZONES
|    └── sample.json <===> A sample zone file including xie.432 DNS records
├── my_dns.py
```

Usage: python3 my_dns.py [server_ip_address] [server_listening_port_number]

Example

```
python3 my_dns.py 127.0.0.1 5300

# To test it, use shell command "dig xie.432 -p 5300 @127.0.0.1"
# Sample output
# ; <<>> DiG 9.10.6 <<>> xie.432 -p 5300 @127.0.0.1
# ;; global options: +cmd
# ;; Got answer:
# ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 11477
# ;; flags: qr aa; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 0

# ;; QUESTION SECTION:
# ;xie.432.         IN  A

# ;; ANSWER SECTION:
# xie.432.      400 IN  A   127.0.0.1
# xie.432.      400 IN  A   127.0.0.1
# xie.432.      400 IN  A   127.0.0.1

# ;; Query time: 0 msec
# ;; SERVER: 127.0.0.1#5300(127.0.0.1)
# ;; WHEN: Tue Nov 20 16:50:53 PST 2018
# ;; MSG SIZE  rcvd: 73
```

A web server at www.yourlastname.432.(In this case, www.xie.432.)

The file server can only handle html files as downloading attachments for now; to support other types of files, need to assign different content-type values to the response header.

File Structure

```
.

├── File
|   └── test.html <==> I choose to use a html file as the sample.
├── my_webs.py
```

Usage: python3 my_webs.py [server_ip_address] [server_listening_port_number]

Example

```
 python3 my_webs.py 127.0.0.1 8080
 # To test it, simply open a browser to access http://127.0.0.1:8080/test.html; a file named "test.html" would then be downloaded.
 # In sum, following all the examples in the my_dns.py and my_webs.py above, if you set your DNS server as 127.0.0.1#5300(I would recommand you to leave the port number as 53 though since most OS does not support a custom DNS port number), you should be able to download the test file with URL http://www.xie.432/test.html.
```