

# Project #3: HTTP Server with Authentication Report

## Team Details

Sharif Haason (ssh128)

## Collaboration

I consulted my own notes on HTTP along with carefully reading both the instructions PDF and the template code. I also browsed my other implementations of TCP sockets in Python to refresh my mind on how sockets work in Python.

## Nonworking Code

All of the code works completely fine on Python 2. I used this test script to check for each request:

```
#!/bin/bash
# create server process
python3 server.py "$1" & > /dev/null

# basic: no username or password posted, no cookies (login)
curl http://localhost:"$1"

# correct username and password posted, no cookies (success)
curl -d "username=bezos&password=amazon" -c cookies.txt -b cookies.txt
http://localhost:"$1"

# non-existent username posted with password, no cookies (bad credentials)
curl -d "username=some&password=none" http://localhost:"$1"

# existing username posted with bad password, no cookies (bad credentials)
curl -d "username=bezos&password=none" http://localhost:"$1"

# exactly one of username or password posted, other field missing, no cookies (bad
credentials)
curl -d "username=bezos" http://localhost:"$1"

# no username or password posted, valid cookie (success)
curl -c cookies.txt -b cookies.txt http://localhost:"$1"

# non-existent username or bad password for existing username, valid cookie (success)
curl -d "username=some&password=none" -c cookies.txt -b cookies.txt
http://localhost:"$1"

# correct username and password, valid cookie (success)
curl -d "username=bezos&password=amazon" -c cookies.txt -b cookies.txt
http://localhost:"$1"

# correct username and password, invalid cookie (bad credentials)
pkill -2 -f server.py
sleep 60
python3 server.py "$1" & > /dev/null
curl -d "username=bezos&password=amazon" -c cookies.txt -b cookies.txt
http://localhost:"$1"

# logout posted, valid cookie (logout)
curl -d "username=bezos&password=amazon" -c cookies.txt -b cookies.txt
http://localhost:"$1" > /dev/null # regenerate cookie
curl -d "action=logout" -c cookies.txt -b cookies.txt http://localhost:"$1"

# logout posted, invalid cookie (logout)
curl -d "username=bezos&password=amazon" -c cookies.txt -b cookies.txt
http://localhost:"$1" > /dev/null # regenerate cookie
pkill -2 -f server.py
sleep 60
python3 server.py "$1" & > /dev/null
```

```
curl -d "action=logout" -c cookies.txt -b cookies.txt http://localhost:"$1"  
  
pkill -2 -f server.py
```

## Difficulties

One unexpected error was parsing the header back and forth between a number and a string. In general, however, once I understood the high-level process, there were few errors.

## Observations

- HTTP at base is simply parsing particular kinds of data sent over a TCP socket.
- Cookies are also very simply just the passing of headers, there's no magic behind it. However, the cookies only exist on the server I created as long as it's up because the cookies are in memory.