

# Assignment 2

2016310932 배현웅

- Environment

Python version : 3.6.9

Operating system : Ubuntu 18.04.5 LTS

- Evaluation

1. Basic HTTP requests & responses - OK
2. Log-in function - OK
3. Persistent HTTP mode - OK

- How to design assignment code

1. Basic HTTP requests & responses - OK

def main() 에서 server 소켓과 bind, listen(5)를 진행합니다. 그리고 계속 while loop 를 돌면서 새로운 socket 이 establish 되면 Thread 에서 request 를 처리하고 response 를 보내기위해서 con\_Thread 를 통해서 serv\_work()함수를 실행합니다.

Serv\_work 함수에서는 소켓을 통해 받은 data 즉, request 를 request\_data, method, file\_location, version 등을 쪼개어서 파싱을 진행합니다.

Method 가 Get 이라면 요청하는 것이 index.html 인지 다른 사진이나 element 를 가져오기 위해서인지 구분하여 index.html 요청하는 경우인 GET / 경우에는 index.html 을 읽어서 response 에 함께 보내줍니다. 또한 cookie value 인 id,password 에서 max-age = 0 을 설정해서 cookie 값을 초기화시켜줍니다.

만약 cookie.html 을 요청하는 경우는 쿠키가 기록되어 있는지(로그인이 되어있는지) 확인한 이후에 cookie.html 파일을 보내줍니다. 여기서 ID 를 해당 쿠키에 해당되는 ID 로 보내주기 위해서 cookie.html 파일을 읽어서 \$ID 부분을 해당 ID 로 replace 해줍니다.

그리고 그 외에 경우인 사진과 같은 폴더내에 있는 데이터들, 예를 들어 직접 index.html 에 access 하는 경우와 같은 경우는 로그인이 되어있는 지 확인한 후에(request 에 함께 오는 쿠키를 통해 확인) 해당 파일을 읽어서 request 에 응답해줍니다. 로그인이 되어있지 않다면 403 forbidden response 를 하게 됩니다.

2. Log-in function - OK

등록된 회원정보는 aaa/aaa 그리고 hyun/1234 입니다. (ID/PW 순)

메인 페이지에서 ID 와 PW 를 입력하는 영역이 존재합니다. 이 부분에 등록되어있는 ID 와 PW 를 입력하면 index.html 에서 POST Method 를 통하여 ./secret.html 을 요청하게 됩니다. 서버영역에서는 메소드가 post 인지 확인한 이후에 request 를 통해서 넘어오는 ID 와 PW 가 등록되어있는지 확인을 하고 등록되어있지 않으면 403 Forbidden Error 메시지를 전송합니다. 만약 해당된다면 secret.html 와 함께 응답메시지를 보내고 연이어서 secret.html 에 포함되어있는 사진파일을 요청받고 Get method 를 통해서 응답하게 됩니다.

### 3. Persistent HTTP mode - OK

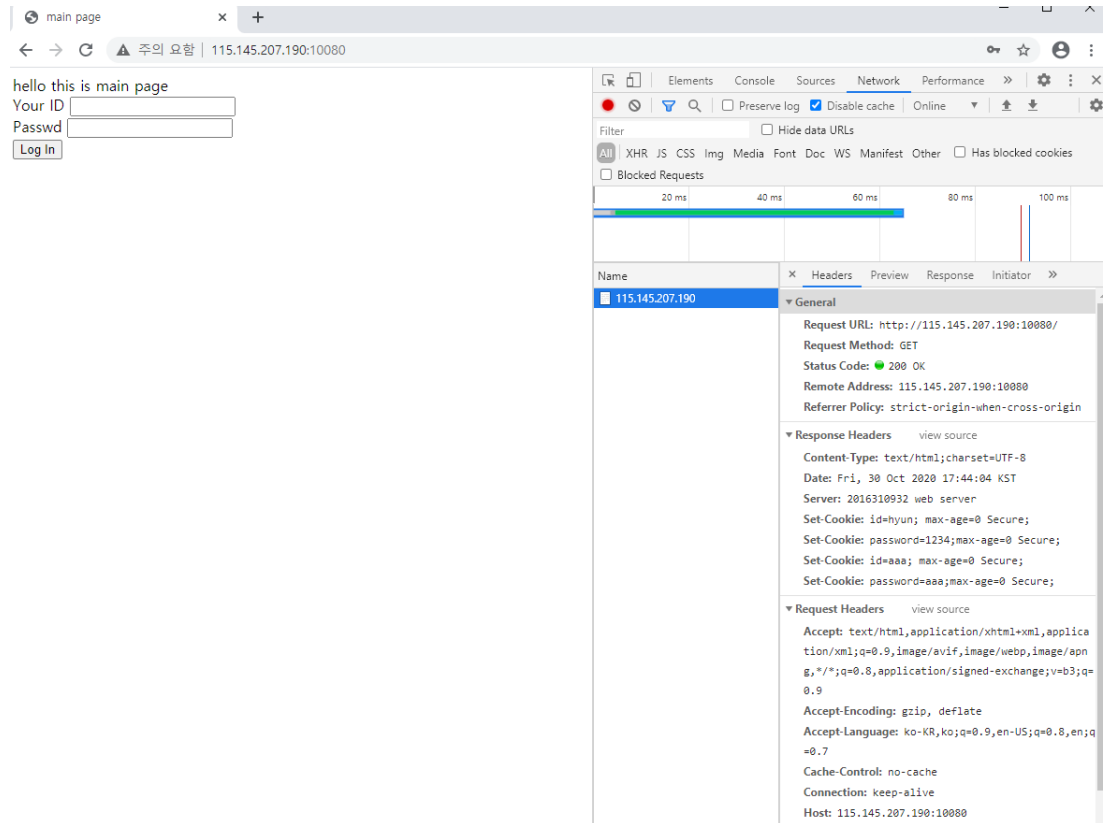
기본적으로 HTTP 1.1 version 에서는 default 로 persistent mode 가 지원합니다. 그리고 header 에

```
"Connection: Keep-Alive\r\nKeep-Alive: timeout=5, max=1000\r\n"
```

응답할 때 같이 붙여서 보내는 것을 통해서 Persistent HTTP mode 를 구현하였습니다.

### 1. Picture for evaluation

### Basic HTTP requests & responses - OK



2.

Figure 1 main page

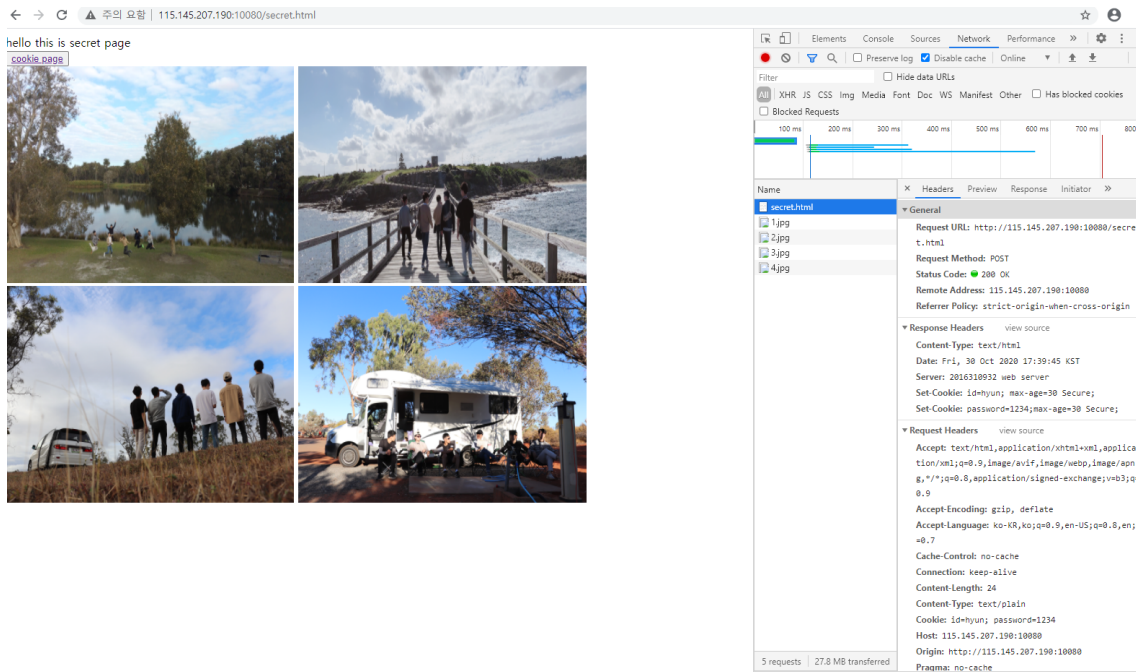


Figure 2 Secret.html with login

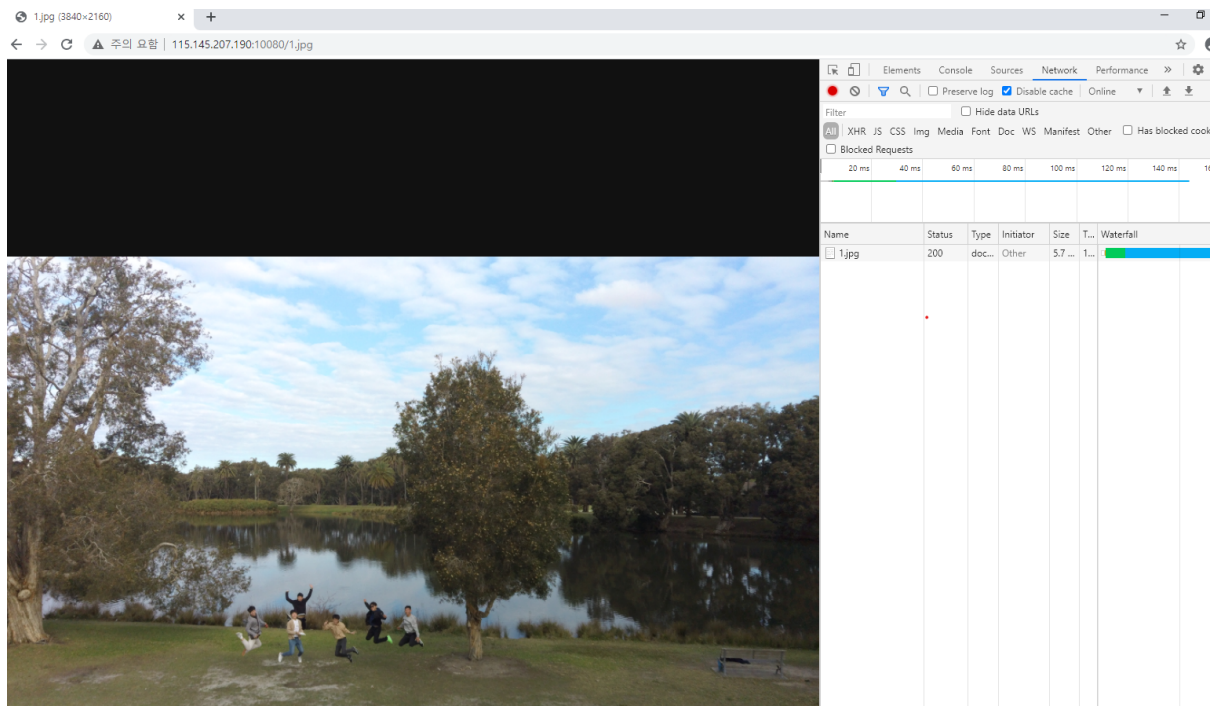


Figure 3 Direct access 1.jpg after log-in

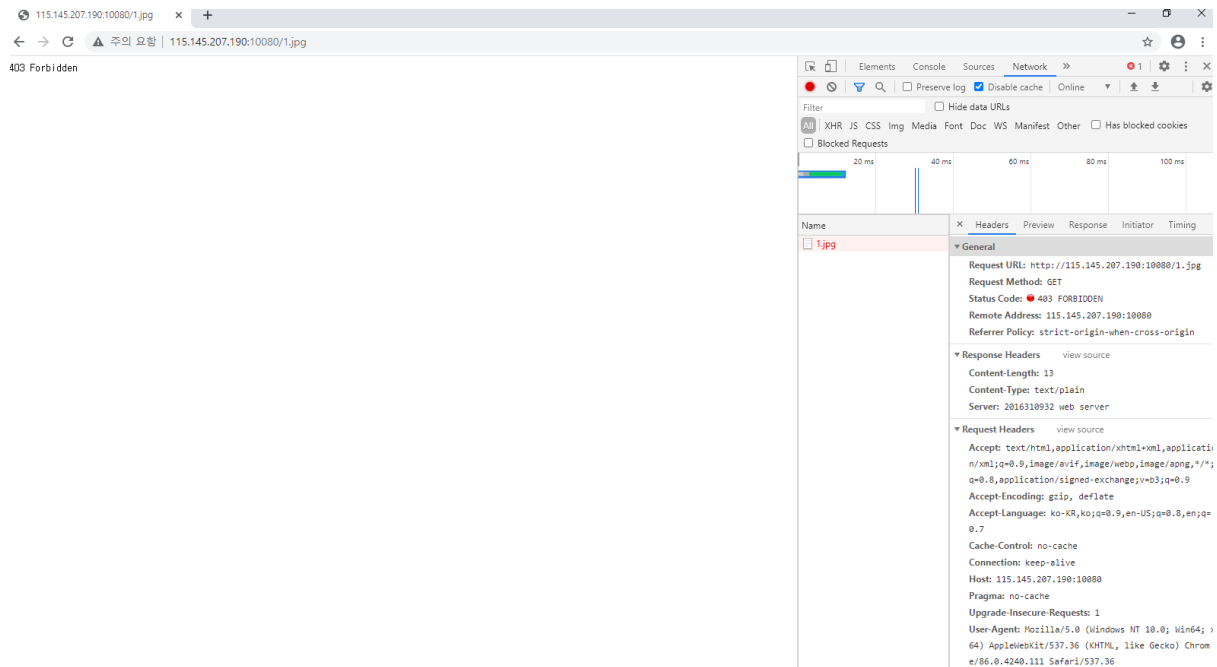


Figure 4 accessing 1.jpg\_after\_cookie\_expired : 403 Forbidden

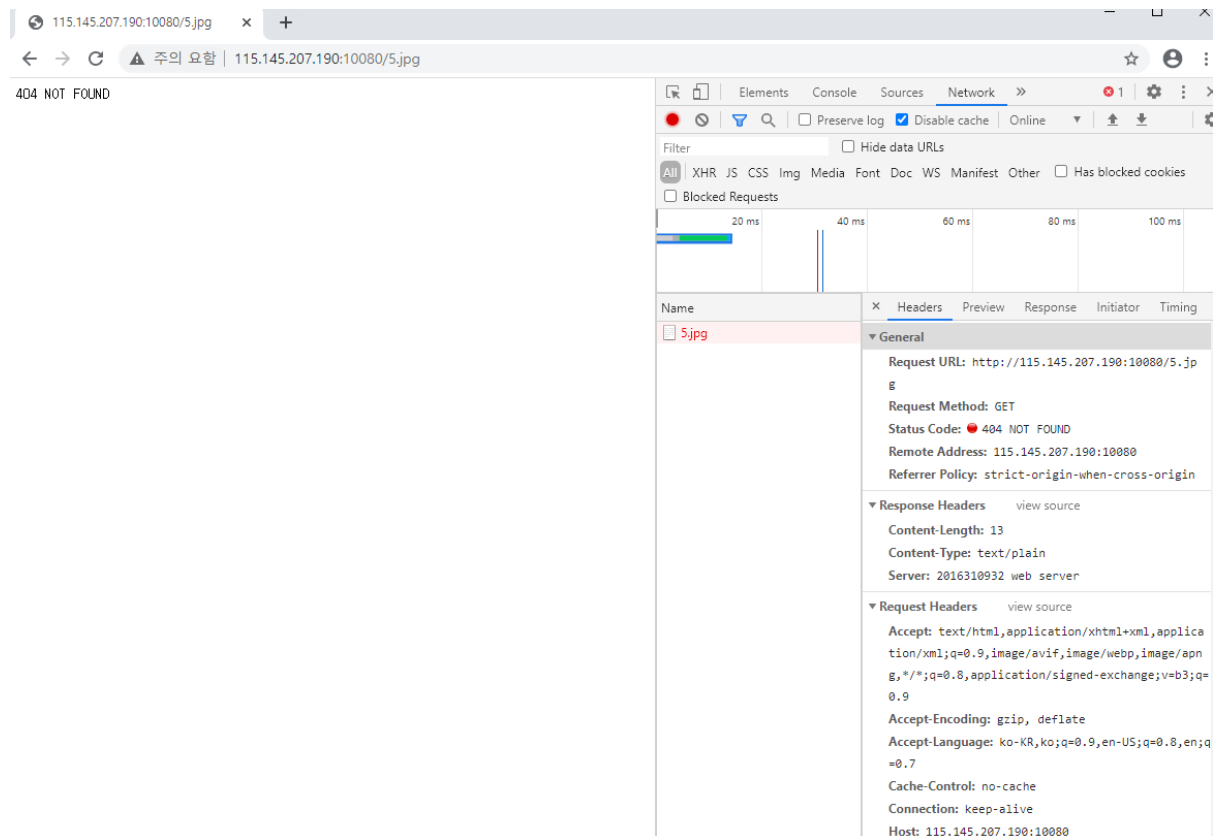
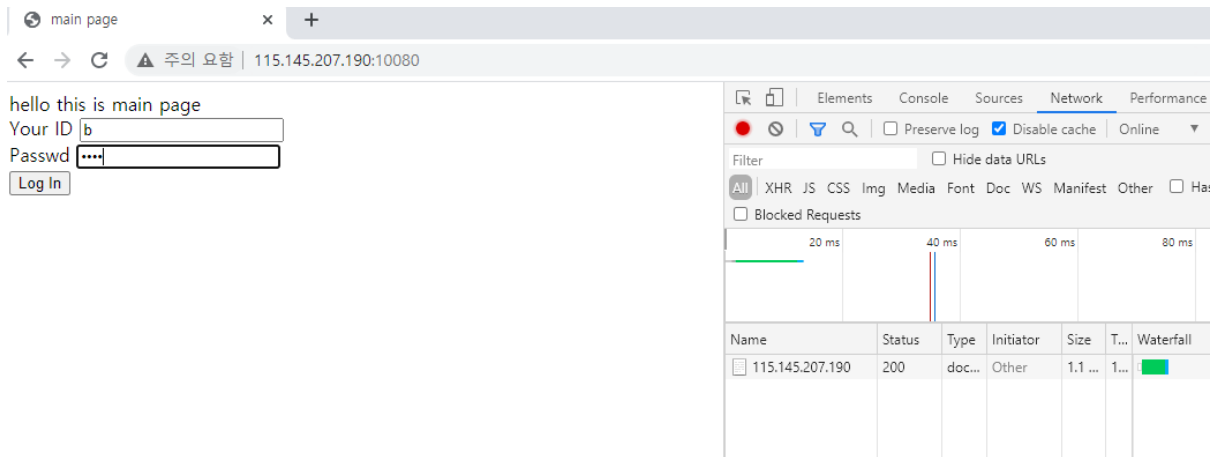
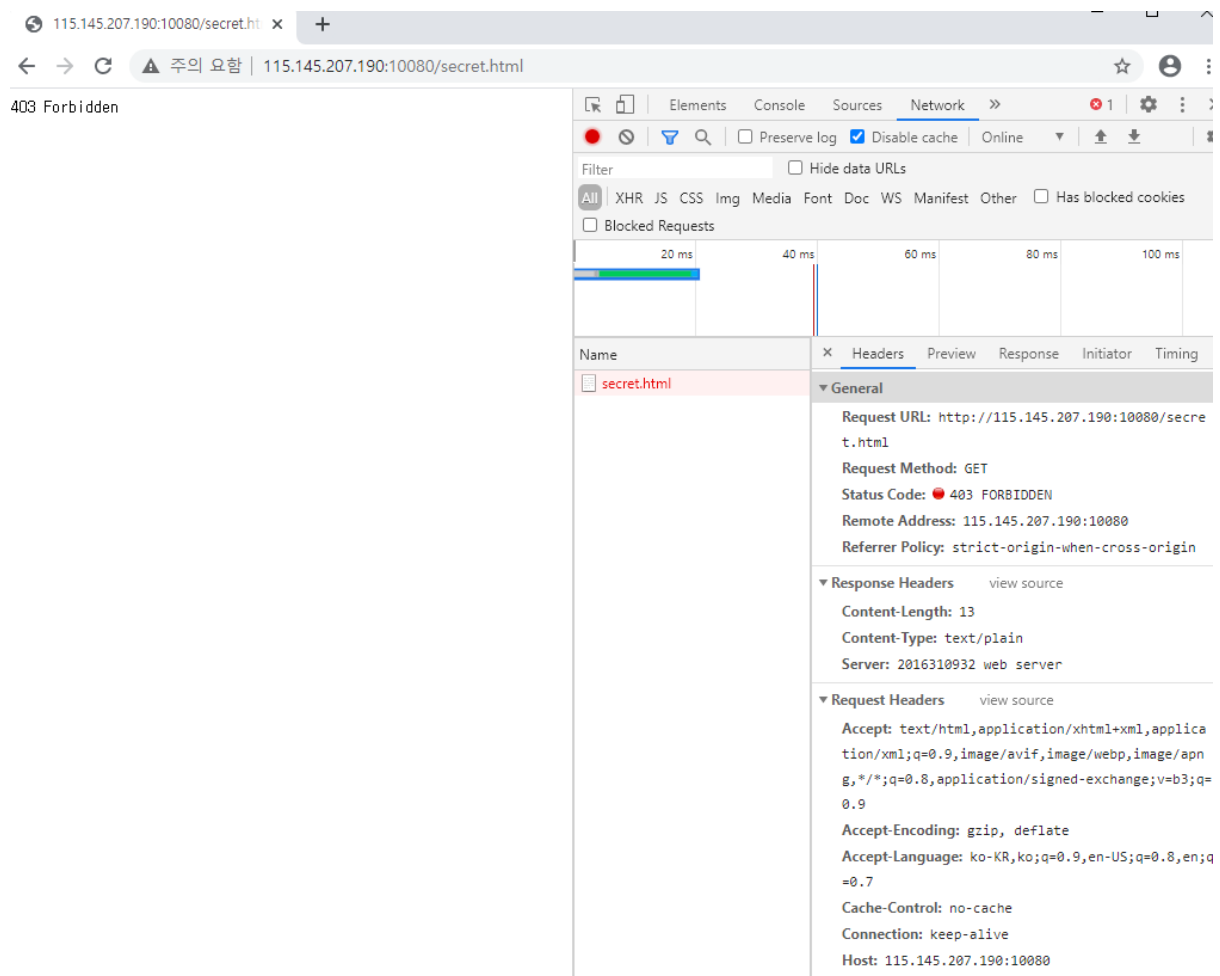


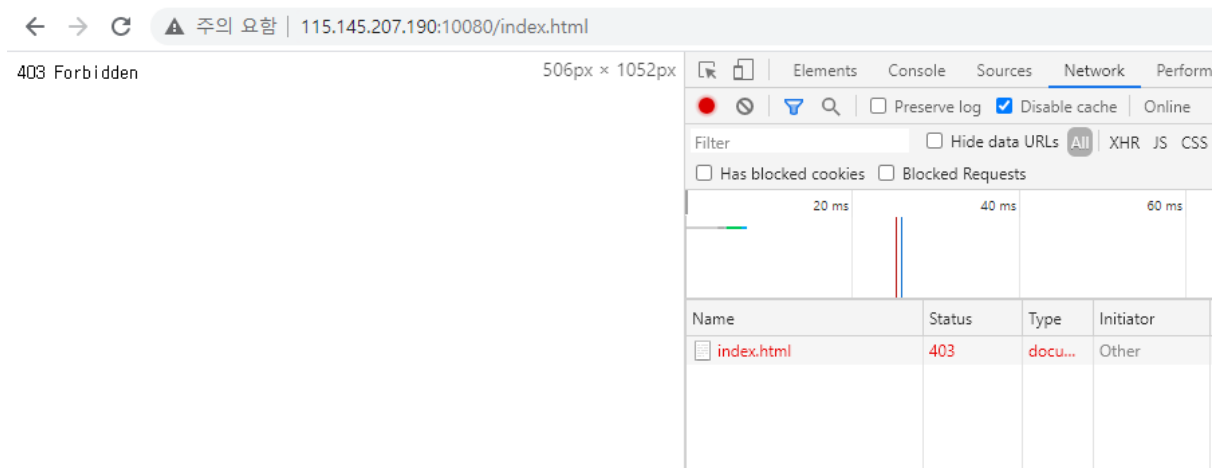
Figure 5 accessing 5.jpg which don't exist : 404 Not Found



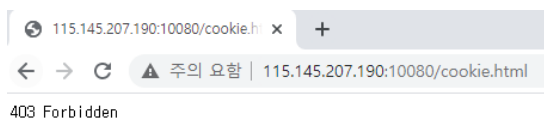
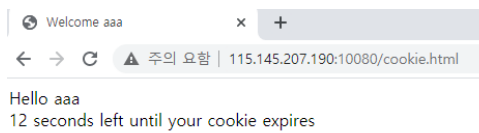
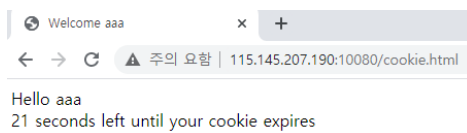
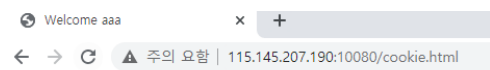
**Figure 6 Try to access invalid ID**



**Figure 7 result of invalid ID**



**Figure 8 Direct index.html without log-in**



**Figure 9 Cache life-time decreasing**

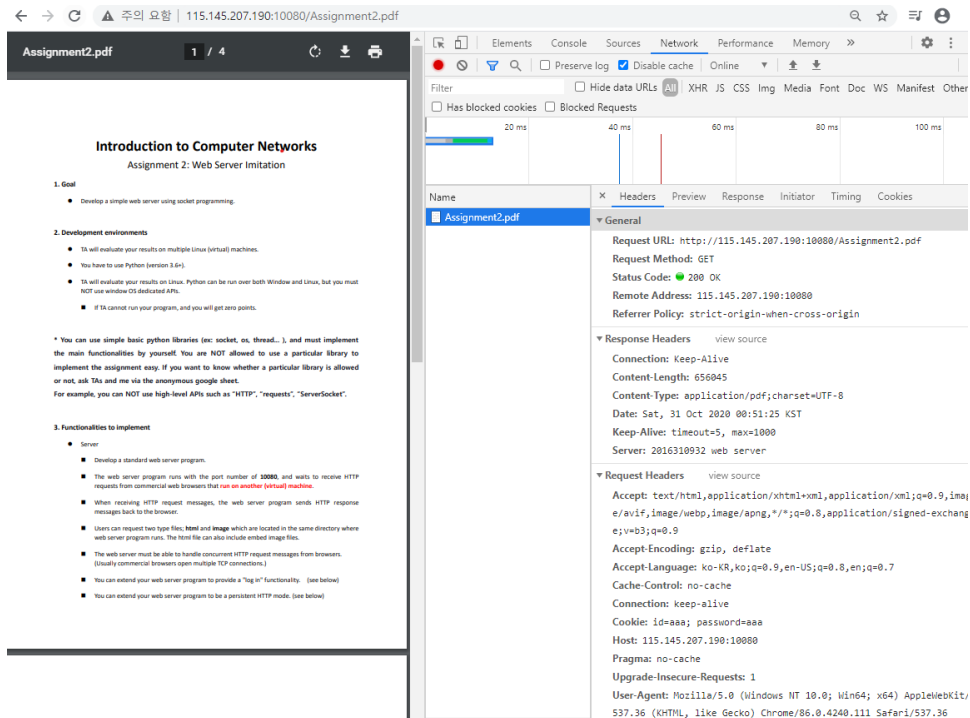


Figure 10 pdf access from Get method

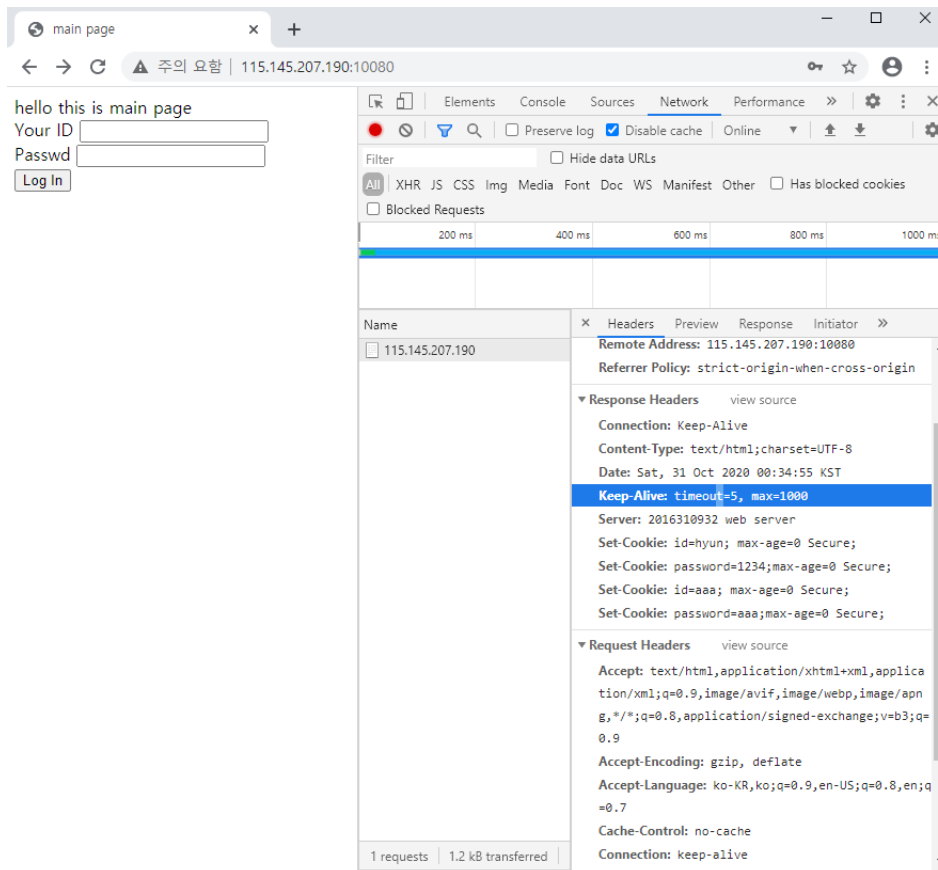


Figure 11 Persistent mod with keep-alive, timeout, max