

컴퓨터 네트워크

-8주차(Phillps Hue)-

학번: 201404377

이름: 진승언

[목표]

TCP socket을 이용한 Phillips Hue제어 HTTP Server 개발을 하는 게 이번 과제의 목표였다. 요약하면 다음과 같다.

1. 서버로 Docker를 사용한다.
2. 전구를 제어할 수 있는 post방식의 form을 담고 있는 Web UI 제작 (html) 한다.
3. 클라이언트에서 GET 요청을 하면 내가 2에서 만든 Web UI 반환해준다.
4. 웹페이지로 POST 요청을 보내고 서버에서는 해당 값을 받아 데이터를 파싱 및 가공하여 브릿지 통신을 하여 Phillips Hue(전구)를 제어한다.

[해결방법]

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<title>Title</title>
<h3>Hue 1 Controller</h3>
<form action="1" method="POST">
<input type="radio" name="light" value="on"> On<br>
<input type="radio" name="light" value="off"> off<br>
<div>Brightness: 0 <input type="range" min="1" max="254" name="brightness" value="1">
254
</div>
X: 0 <input name="color_x" type="number" value="0" step="0.1" max="1" min="0"> 1
</p>
Y: 0 <input name="color_y" type="number" value="0" step="0.1" max="1" min="0"> 1
</p>
<input type="submit" value="Submit">
</form>

<h3>Hue 2 Controller</h3>
<form action="2" method="POST">
<input type="radio" name="light" value="on"> On<br>
<input type="radio" name="light" value="off"> off<br>
<div>Brightness: 0 <input type="range" min="1" max="254" name="brightness" value="1">
254
</div>
X: 0 <input type="number" name="color_x" value="0" step="0.1" max="1" min="0"> 1
</p>
Y: 0 <input type="number" name="color_y" value="0" step="0.1" max="1" min="0"> 1
</p>
<input type="submit" value="Submit">
</form>

<h3>Hue 3 Controller</h3>
<form action="3" method="POST">
<input type="radio" name="light" value="on"> On<br>
<input type="radio" name="light" value="off"> off<br>
<div>Brightness: 0 <input type="range" min="1" max="254" name="brightness" value="1">
254
</div>
X: 0 <input type="number" name="color_x" value="0" step="0.1" max="1" min="0"> 1
</p>
Y: 0 <input type="number" name="color_y" value="0" step="0.1" max="1" min="0"> 1
</p>
<input type="submit" value="Submit">
</form>
</head>
<body>
</body>
```

위는 전구를 제어하는 웹페이지인 hueController.html 파일이다. (실수로 head태그에 다 작성했다) form 태그는 post 방식으로 하였고 전원 on/off, 밝기 그리고 x, y 컬러값을 설정할 수 있다. 그리고 총 3개의 전구를 제어할 수 있다.

The screenshot shows a web browser window with the address bar displaying 'localhost:1234/hueController.html'. The page contains three identical sections, each titled 'Hue 1 Controller', 'Hue 2 Controller', and 'Hue 3 Controller' respectively. Each section includes a radio button group with 'On' and 'off' options, a 'Brightness: 0' label followed by a slider bar ending at '254', and two input fields labeled 'X: 0' and 'Y: 0' with a range from '0' to '1'. A 'Submit' button is located below each set of controls.

UI는 위와 같다.

서버단 코드는 다음과 같다.

```
from phue import Bridge
```

pip3 install phue 로 설치 후 위와 같이 import 해준다. 필립스 휴 와 브릿지 통신을 하기 위해서이다.

```

def accept_client(s_socket):
    while True:
        print("wait ... ")
        conn, addr = s_socket.accept()
        print(conn)
        print(addr)
        data = conn.recv(5000).decode('utf-8')
        #data example GET /index.html /HTTP1.1...content-type...
        print("data ==> " + data)
        #http_method is POST or GET
        http_method = data.split(" ")[0]

        try:
            if http_method == "GET":
                #http_dir example /index.html
                http_dir = data.split(' ')[1]
                print('http_dir(GET) ==> '+http_dir)
                #remove / and open that html file
                f = open(http_dir[1:])
                outputdata = f.read()
                print("connected by ", end='')
                print(conn.getpeername())
                print("connect client... ", end='')
                print(s_socket.getsockname()[1])
                print("IP:{0}, data:{1}".format(addr[0], data))
                print(outputdata)
                conn.send('HTTP/1.1 200 OK\r\n'.encode('utf-8'))
                conn.send('Content-Type: text/html\r\n\r\n'.encode('utf-8'))
                conn.send(outputdata.encode('utf-8'))
                conn.close()
            elif http_method == "POST":
                http_dir = data.split(' ')[1]

```

서버소켓을 연 후 요청을 기다린다. 요청이 오면 HTTP 헤더를 파싱해서 GET 요청인지 POST 요청인지 분기해준다.

GET 요청이라면 요청한 경로를 파싱해 해당 html을 클라이언트에게 응답해준다.

```

conn.close()
elif http_method == "POST":
    http_dir = data.split(' ')[1]
    print("http_dir(POST) ==> " + http_dir)
    f = open("hueController.html")
    #parsing post data ex) light=on&brightness=1&color_x=0&color_y=0
    post_data = data.split('\r\n\r\n')[1]
    print(post_data)
    #parsing post value (ex on, 1)
    hue_num = http_dir[1:]
    print(hue_num)
    power = post_data.split('&')[0].split('=')[1]
    print(power)
    brightness = post_data.split('&')[1].split('=')[1]
    print(brightness)
    color_x = post_data.split('&')[2].split('=')[1]
    print(color_x)
    color_y = post_data.split('&')[3].split('=')[1]
    print(color_y)

    #serving page
    outputdata = f.read()
    conn.send('HTTP/1.1 200 OK\r\n'.encode('utf-8'))
    conn.send('Content-Type: text/html\r\n\r\n'.encode('utf-8'))
    conn.send(outputdata.encode('utf-8'))
    conn.close()

    #connect bridge

    bridge = Bridge('192.168.0.209')
    print("connect!!!!!!!!!!!!")
    bridge.connect()
    lights = bridge.lights
    print(lights)
    if power == "on":
        lights[int(hue_num)-1].on = True
    else:
        lights[int(hue_num)-1].on = False

    lights[int(hue_num)-1].brightness = int(brightness)
    lights[int(hue_num)-1].xy = [float(color_x), float(color_y)]

```

클라이언트에서 서빙된 웹페이지 post 방식의 form 태그를 통해 post 요청을 하면 서버에서 위와 같이 받는다. 전달된 데이터를 파싱하여 전구의 번호(hue_num), 전원(power), 밝기(brightness), 컬러값(color_x, color_y)에 저장한다. 그리고 Bridge를 만들고 연결하여 받은 값을 이용해 전구를 제어한다. lights 안에는 휴 전구 리스트가 들어있다.

```

except Exception as ex:
    print('exception state')
    f = open('noIndex.html')
    outputdata = f.read()
    print("connected by ", end='')
    print(conn.getpeername())
    print("connect client... ", end='')
    print(s_socket.getsockname()[1])
    print("IP:{0}, data:{1} ".format(addr[0], data))
    print(outputdata)
    conn.send('HTTP/1.1 404 Not Found\r\n'.encode('utf-8'))
    conn.send('Content-Type: text/html\r\n\r\n'.encode('utf-8'))
    conn.send(outputdata.encode('utf-8'))
    conn.close()

```

만약 잘못된 경로로 요청하거나 예외가 발생하면 noIndex.html을 클라이언트에게 보내주었다.

실행 영상이다.

https://www.youtube.com/watch?v=mUazpV_kJL8

[과제후기]

휴 와 와이파이 연결 에러가 많아서 고생했다. 그래도 한 번에 통신이 되어서 다행이다.