

# Web Server Project

2016003409 김원배

## 1. Request Message 각 field에 해당하는 값 (참조 : RFC2616)

```
Wonui-MacBookAir:tcp-server wonbae$ ./server 8000
GET /index.html HTTP/1.1
Host: localhost:8000
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_4) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/13.1 Safari/605.1.15
Accept-Language: ko-kr
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

Safari를 이용해, server에 접속했을 때, server로 전달되는 request message를 출력한 것이다.

http/1.1프로토콜에서의 Full Request는

```
Request          = Request-Line           ;
                   *( ( general-header     ;
                       | request-header    ;
                       | entity-header ) CRLF ) ;
                   CRLF
                   [ message-body ]
```

의 형태로 이루어진다.

이 중, GET /index.html HTTP/1.1은 request-line에 해당하고. 이 후에, 나오는 field들을 header field에 해당한다.

더 세부적으로, Upgrade, Connection은 general-header field,

Host, accept, user-agent는 request-header field에 해당한다.

## 2. 대략적인 서버 디자인

소켓 생성 -> IP 주소와 PORT 번호 할당 -> 연결 요청 가능 상태로 변경

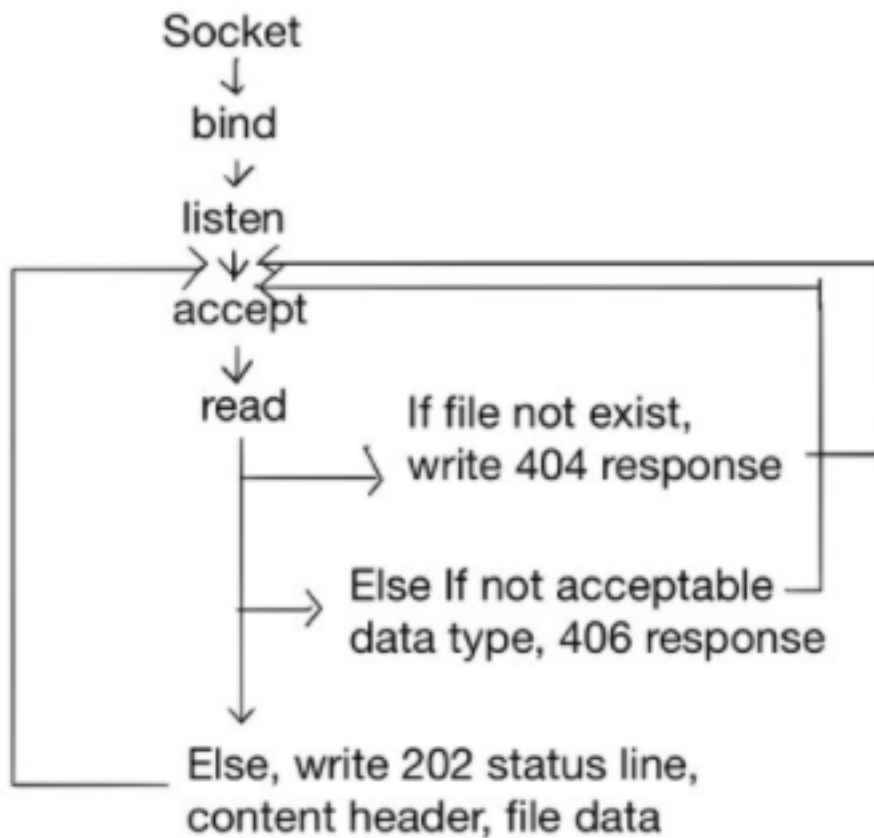
-> 연결 요청 수락 (루프문 시작) -> response message 분석 (http version, file name ... )

-> file 존재 여부 확인 (file 없을 시, 404 not found를 담은 response message를 클라이언트로 보내고, 연결 요청을 받는 루프문으로 돌아감.) -> file의 확장자 확인 (서버에서

처리할 수 없는 확장자일 경우, 406 not acceptable을 담은 response message를 클라이언트로 보내고, 연결 요청을 받는 루프문으로 돌아감) -> file을 버퍼에 담고, 202 ok와

content type, length header를 담은 response message를 보내고, file을 담은 버퍼를 클라이언트로 보내고, 연결 요청을 받는 루프문으로 돌아감)

이엔트로 보내줌 -> 다시 루프.



### 3. 구현 시 어려웠던 점 / 해결방법

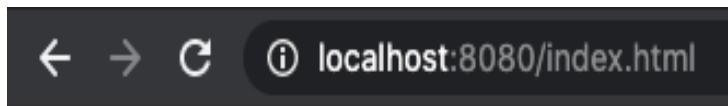
1) Request message와 response message의 기본적인 구조를 몰랐기 때문에, 어떤 부분을 가져와 처리하고, response message에 담아 주어야 하는지 막막했습니다. 하지만, RFC 문서에 관련 정보들이 자세하게 적혀 있었고, RFC 문서에서 잘 알지 못했던 부분을 공부하고, 기본적인 틀에 대해 이해하게 되었습니다.

2) 가장 어려움을 겪었던 부분은, C언어에서 char\*를 다루는 부분이었는데, 많은 시도를 해보고, sprintf 함수를 적절히 사용해, 관련 어려움을 해결할 수 있었습니다. C언어로 file을 다루어본 적도 없었기 때문에, 이 것도 별도로 공부해야 했습니다.

3) mac os환경에서 코드를 작성한 후, ubuntu 환경에서 에러가 발생해, 다시 ubuntu 환경에서 관련 에러를 확인해, 두 os에서 정상 작동하도록 수정해주었습니다.

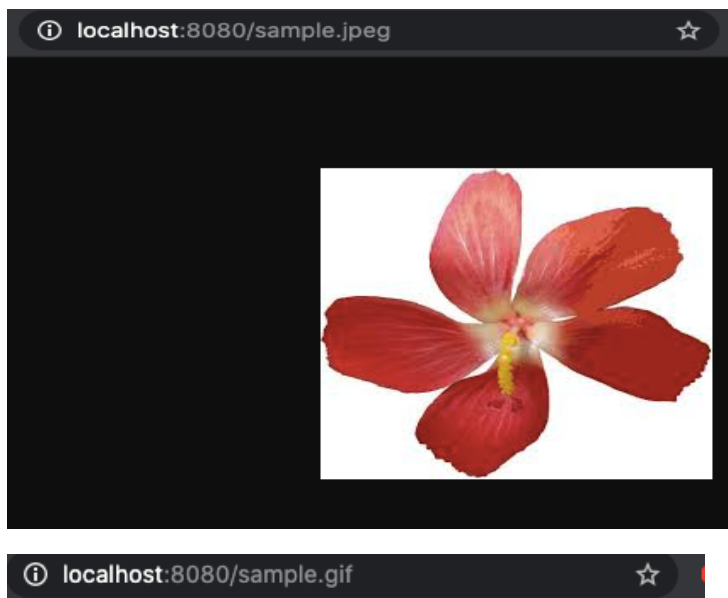
#### 4. 동작 예시

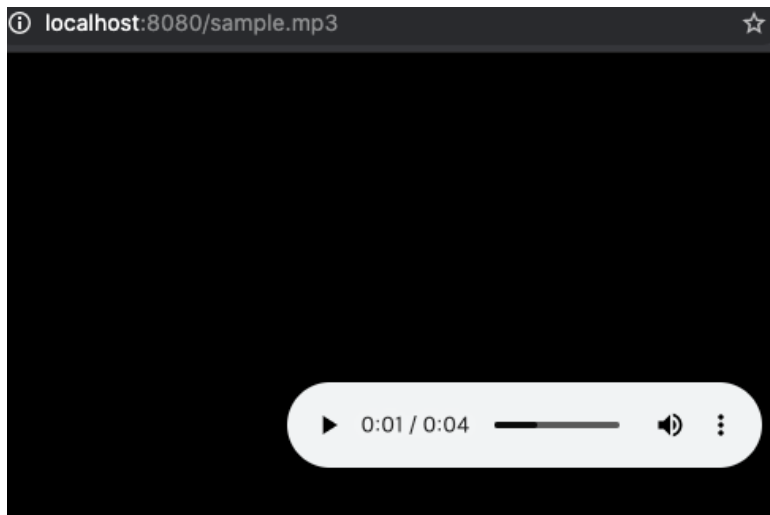
- 1) 존재하는 파일, 서버에서 처리할 수 있는 확장자



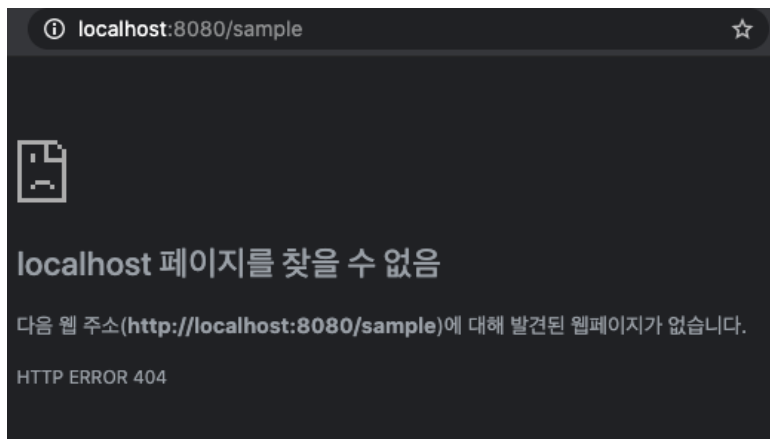
Hello

- [image file](#)
- [motion picture](#)
- [document](#)
- [music](#)





2) 존재하지 않는 파일을 요청했을 때



3) 존재하지만, 처리할 수 없는 확장자 파일을 요청했을 때

