컴퓨터 네트워크

TCP 기반 Socket Programming
HTTP Request / Response

국민대학교 소프트웨어학부
20162347 김창규

소스 파일 및 동작 환경

[개발 환경]

운영체제	MS Windows 11 (10↑)
작성언어	Python 3.7.9
IDE	VS Code

[Python Module]

socket
datetime

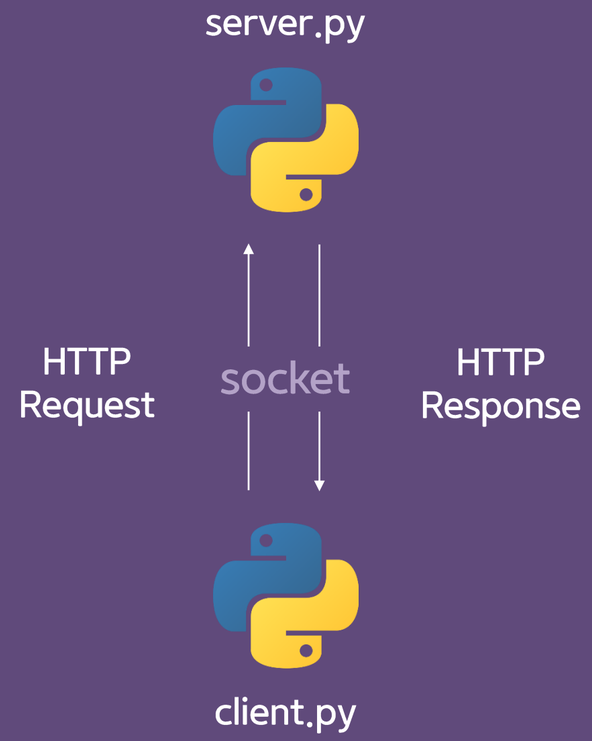
[실행 환경]

Server	Labtop
Client	Labtop Postman

[디렉토리 구조]

```
src/
 └  server.py
    (서버 사이드 프로그램)
    client.py
    (클라이언트 사이드 프로그램)
.gitattributes    (깃허브 파일)
.gitignore        (깃허브 파일)
README.md         (사용법)
```

[동작 흐름]



실행 방법 및 가이드

서버 실행

python server.py

클라이언트 실행

python client.py

클라이언트 요청

METHOD >>
GET | POST | DELETE
(외에는 405 Error)
URL >>
/
(URL 중첩은 404 Error)
VERSION >>
HTTP/1.1
(상관 없음)
METHOD >>
서버로 전송할 메시지
(POST 인 경우에만 가능)

HTTP Request / Response Message

Request

Header

[METHOD] [URL] [VERSION]
Host: HOST:PORT
Content-Type: text/plain
User-Agent: CLIENT_NAME
Connection: keep-alive

Response

Header

[VERSION] [STATUS_CODE]
Server: SERVER_NAME
Date: DATETIME
Content-Type: text/plain
Connection: keep-alive

Body

...
TEXT_MESSAGE
...

Request / Response Message의 Header 필드는 최대한 간단하게 구성

Postman을 활용한 요청에서 Socket.close()을 하지 않으면 다음 요청을 처리하지 못하는 오류가 있어 매번 연결-처리-해제 하도록 코드 작성

소스 파일 설명

client.py

```
from socket import *

HOST, PORT = "127.0.0.1", 80          # 호스트 IP, PORT 설정
USER_AGENT = "..."                    # 클라이언트 AGENT 이름

while True:
        # TCP Socket 생성
        CLIENT_SOCKET = socket(AF_INET, SOCK_STREAM)
        # TCP Connection
        CLIENT_SOCKET.connect((HOST, PORT))

        ...

        # HTTP Request Message 송신
        CLIENT_SOCKET.sendall(REQUEST.encode())

        # HTTP Response Message 수신
        RAW_RESPONSE = CLIENT_SOCKET.recv(65535)
        print(RAW_RESPONSE.decode())

        CLIENT_SOCKET.close()    # TCP 연결 해제
```

server.py

```
from socket import *
from datetime import datetime

MESSAGE = []                          # 메시지 저장 변수
HOST, PORT = "127.0.0.1", 80          # 호스트 IP, PORT 설정
SERVER_NAME = "..."                   # 서버 AGENT 이름

# TCP Socket 생성/바인딩/TCP_CONNECTION 설정
SERVER_SOCKET = socket(AF_INET, SOCK_STREAM)
SERVER _SOCKET.bind((HOST, PORT))
SERVER _SOCKET.listen(0)

while True:
        # Client 연결 대기
        CLIENT_SOCKET, ADDR = SERVER_SOCKET.accept()

        # HTTP Request Message 수신
        RAW_REQUEST = CLIENT_SOCKET.recv(65535)

        ...METHOD PROCESSING...

        # HTTP Response Message 송신
        CLIENT_SOCKET.sendall(RESPONSE.encode())

CLIENT_SOCKET.close()                 # TCP 연결 해제 (Client)
SERVER_SOCKET.close()                 # TCP 소켓 해제 (Server)
```

(1) GET Method - 200 OK

정상적으로 GET을 요청한 경우

- URL == /

  (요청 URL이 루트)

  서버 메모리에 저장 중인 메시지를 읽어옵니다.

클라이언트 사이드 프로그램

```
C:\Users\kimc9\Desktop\Github\ComputerNetwork\src>python client.py
METHOD >> GET
URL >> /
HTTP VERSION >> HTTP/1.1
HTTP/1.1 200 OK
Server: Python Socket Program Server
Date: Mon, 02 May 2022 17:33:51 KST
Content-Type: text/plain
Connection: keep-alive

HTTP GET Method (Success)
```

서버 사이드 프로그램

```
C:\Users\kimc9\Desktop\Github\ComputerNetwork\src>python server.py
GET / HTTP/1.1
Host: 127.0.0.1:80
Content-Type: text/plain
User-Agent: Python Socket Program Client
Connection: keep-alive
```

(2) GET Method - 404 Not Found

GET을 잘못 요청한 경우

- URL == /wrong/

(요청 URL이 잘못된 경우)

HTTP GET Method (Failed) 메시지 반환

클라이언트 사이드 프로그램

```
METHOD >> GET
URL >> /wrong/
HTTP VERSION >> HTTP/1.1
HTTP/1.1 404 Not Found
Server: Python Socket Program Server
Date: Mon, 02 May 2022 17:49:46 KST
Content-Type: text/plain
Connection: keep-alive

HTTP GET Method (Failed)
```

서버 사이드 프로그램

```
GET /wrong/ HTTP/1.1
Host: 127.0.0.1:80
Content-Type: text/plain
User-Agent: Python Socket Program Client
Connection: keep-alive
```

(3) POST Method - 201 Created

정상적으로 POST를 요청한 경우

- URL == /

  (요청 URL이 루트)

  MESSAGE = ...

  (서버 메모리에 저장할 메시지)

  서버 메모리에 메시지를 작성(추가)합니다.

클라이언트 사이드 프로그램

```
METHOD >> POST
URL >> /
HTTP VERSION >> HTTP/1.1
MESSAGE >> Computer Network
HTTP/1.1 201 Created
Server: Python Socket Program Server
Date: Mon, 02 May 2022 17:56:18 KST
Content-Type: text/plain
Connection: keep-alive

HTTP POST Method (Success)

Computer Network
```

서버 사이드 프로그램

```
POST / HTTP/1.1
Host: 127.0.0.1:80
Content-Type: text/plain
User-Agent: Python Socket Program Client
Connection: keep-alive

Computer Network
```

(4) POST Method - 404 Not Found

POST를 잘못 요청한 경우

- URL == /wrong/

  (요청 URL이 잘못된 경우)

HTTP POST Method (Failed) 메시지 반환

클라이언트 사이드 프로그램

```
METHOD >> POST
URL >> /wrong/
HTTP VERSION >> HTTP/1.1
MESSAGE >> Computer Network
HTTP/1.1 404 Not Found
Server: Python Socket Program Server
Date: Mon, 02 May 2022 18:01:55 KST
Content-Type: text/plain
Connection: keep-alive

HTTP POST Method (Failed)
```

서버 사이드 프로그램

```
POST /wrong/ HTTP/1.1
Host: 127.0.0.1:80
Content-Type: text/plain
User-Agent: Python Socket Program Client
Connection: keep-alive
```

(5) DELETE Method - 200 OK

정상적으로 DELETE를 요청한 경우

- 조건 없음

  (HOST:PORT로 DELETE 오면 무조건 수행)

  서버 메모리에 저장 중인 메시지를 비웁니다.

클라이언트 사이드 프로그램

```
METHOD >> DELETE
URL >> /
HTTP VERSION >> HTTP/1.1
HTTP/1.1 200 OK
Server: Python Socket Program Server
Date: Mon, 02 May 2022 18:09:16 KST
Content-Type: text/plain
Connection: keep-alive

HTTP DELETE Method (Success)
```

서버 사이드 프로그램

```
DELETE / HTTP/1.1
Host: 127.0.0.1:80
Content-Type: text/plain
User-Agent: Python Socket Program Client
Connection: keep-alive
```

(6) PUT Method - 405 Method Not Allowed

허가되지 않은 요청을 한 경우

- METHOD == PUT

(메소드가 잘못된 경우)

HTTP [메소드] Method (Failed) 메시지 반환
+ GET/POST/DELETE 외 전부

클라이언트 사이드 프로그램

```
METHOD >> PUT
URL >> /
HTTP VERSION >> HTTP/1.1
HTTP/1.1 405 Method Not Allowed
Server: Python Socket Program Server
Date: Mon, 02 May 2022 18:13:54 KST
Content-Type: text/plain
Connection: keep-alive

HTTP PUT Method (Failed)
```

서버 사이드 프로그램

```
PUT / HTTP/1.1
Host: 127.0.0.1:80
Content-Type: text/plain
User-Agent: Python Socket Program Client
Connection: keep-alive
```

(ADD) Postman으로 서버에 요청

Postman

Response Headers

KEY	VALUE
Server	Python Socket Program Server
Date	Mon, 02 May 2022 18:19:08 KST
Content-Type	text/plain
Connection	keep-alive

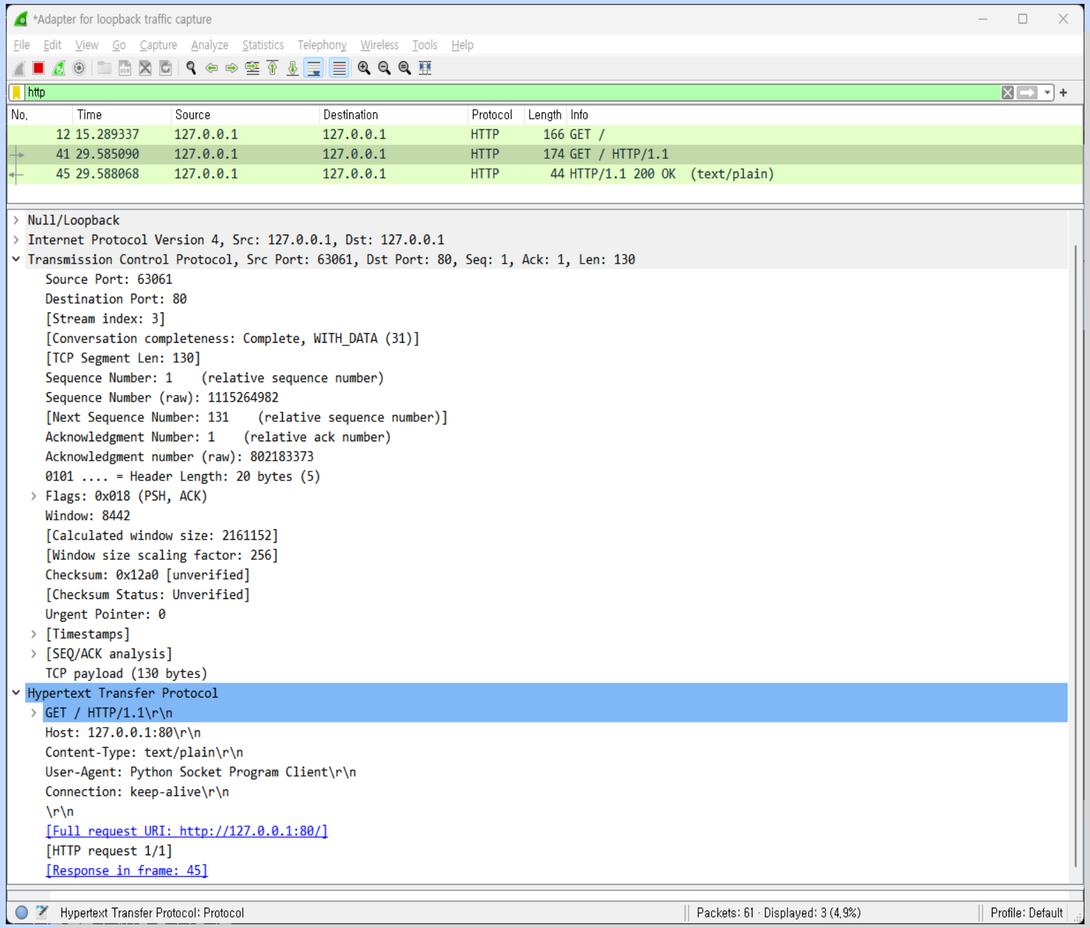
Response Body

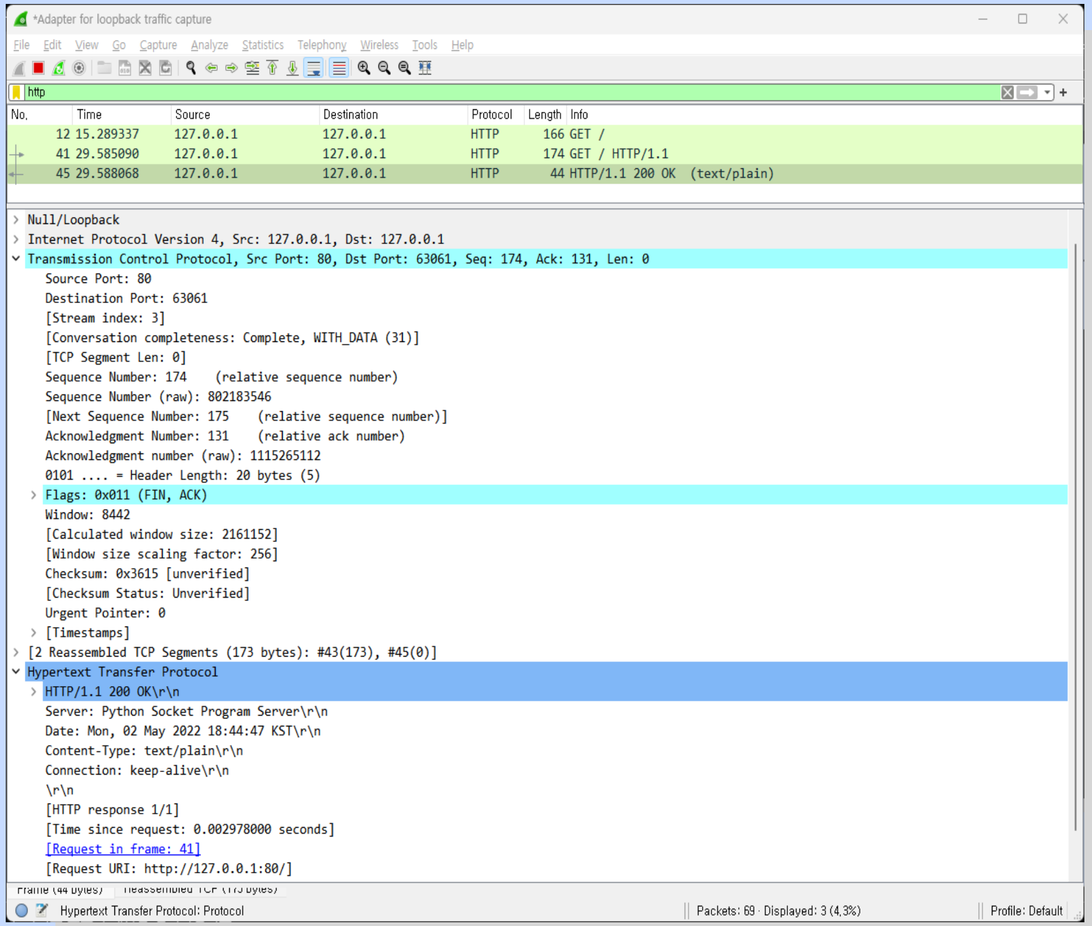
```
1   HTTP GET Method (Success)
2
3
```

서버 사이드 프로그램

```
C:\Users\kimc9\Desktop\Github\ComputerNetwork\src>python server.py
GET / HTTP/1.1
User-Agent: PostmanRuntime/7.29.0
Cache-Control: no-cache
Postman-Token: 21d2472e-624a-4386-8f45-ac9bb475d846
Host: 127.0.0.1:80
Connection: keep-alive
```

(ADD) Wireshark 캡쳐 (1)





(ADD) Wireshark 캡쳐 (2)