**Project Example #2,1**: Building a webclient

**Name**: Kwon DoHyun

**Student ID**: 2023065350

**Organization**: Department of computer science and engineering, Hanyang Univ. (Seoul, Republic of Korea)

**Introduction**

**This assignment implements a web client using socket programming. In the process, you will learn how to handle HTTP requests and responses. You will also learn how the HTTP protocol, which is the foundation of the web, works.**

**Contents of the attachment**

**WebClient.java**

|  |
| --- |
| import java.io.\*;  import java.net.\*;  /\*\*  \* WebClient class implements a simple web client.  \* Its primary responsibilities include:  \* 1. Initializing the tcp connection to web server  \* 2. send HTTP request and receive HTTP response  \*/  public class WebClient {  public static void main(String[] args) {  // Set the host, port and resource to send HTTP Request  String host = "example.com";  int port = 80;  String resource = "/index.html";  // Mission 1: Establish a socket connection to Server  try (  //Fill #1, Set TCP socket to HTTP Web Server  Socket socket = new Socket(host, port);  //Fill #2, create PrintWirter instance with socket’s OutputStream  PrintWriter out = new PrintWriter(socket.getOutputStream(), true);  //Fill #3, Get input stream from server, and insert it to BufferedReader instance  BufferedReader in new BufferedReader(new InputStreamReader(socket.getInputStream()))) {  /\*\*  \* Improve your HTTP Client to provide other request Methods(POST, DELETE, …)  \* and also improve to handle headers(Content-Type, User-Agent, …)  \*/  // Mission 2: Send HTTP GET Request and Read and display the response  // Fill#4, Send HTTP GET request  out.println("HTTP GET: " + resource);  out.println("host: " + host);  out.println("connect: close");  out.println();  // Mission 3: Read and display the response  //Fill#5, Read and display the response  String responseLine;  while ((responseLine = in.readLine()) != null) {  System.out.println(responseLine);  }  } catch (IOException e) {  e.printStackTrace();  }  }  } |

**Result**

텍스트, 스크린샷, 폰트, 문서이(가) 표시된 사진

자동 생성된 설명

**Block diagram of UDP Web Server & Web Client**

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

**Flow diagram of UDP Web Server & Web Client**

텍스트, 도표, 라인, 폰트이(가) 표시된 사진

자동 생성된 설명

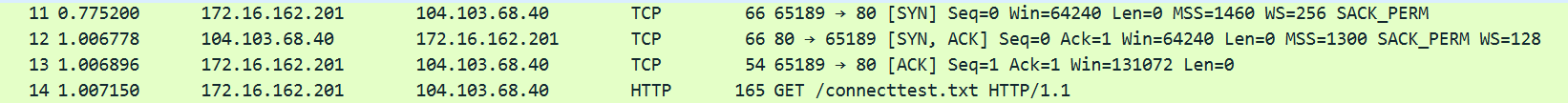
**Instructions**: How to run the program

1. Run Web Server
2. Run Web Client
3. Web client receives response and prints HTTP Request

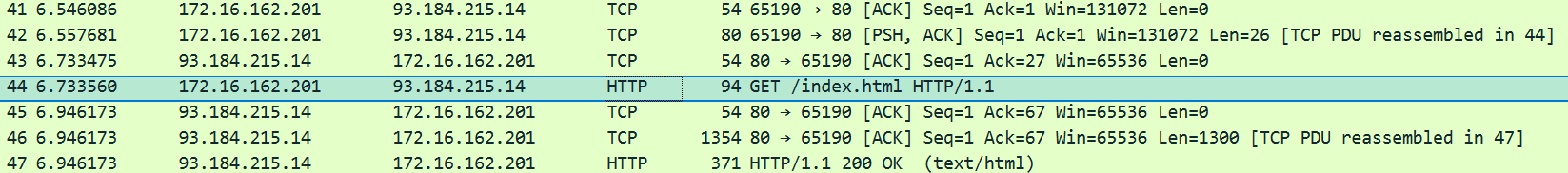
**Wireshark**

**172.16.162.201: Client IP, 104.103.68.40: Server IP**

**Client HTTP Request**



If you look at packet number 11 in the picture above, you can see that a SYN packet is sent to the server, and if you look at packet number 14, you can see that an HTTP GET request is sent to the server.



If you look at packet number 44 in the picture above, you can see that an HTTP GET request is sent to another server (93.184.215.14). If you look at packet number 47, the client receives a response of HTTP/1.1 200 ok.

텍스트, 전자제품, 스크린샷, 디스플레이이(가) 표시된 사진

자동 생성된 설명

**How the program works**

1. Set the server address, port, etc.
2. Connect the Web Server and Socket.
3. Send an HTTP GET request to the Web Server.
4. Receive and output the Server response.
5. Terminate the connection.