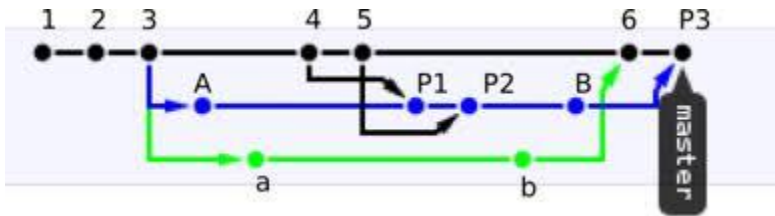


# WebServer 구현





## webserver\_step1

### ServerSocket 생성

```

public class WebServer {
    private final static Logger log = Logger.getLogger(WebServer.class.getName());

    public static void main(String argv[]) throws Exception {
        // 서버소켓을 생성한다. 웹서버는 기본적으로 80번 포트를 사용한다.
        ServerSocket listenSocket = new ServerSocket(80);
        log.log(Level.INFO, "WebServer Socket Created");

        // 클라이언트가 연결될 때까지 대기한다.
        Socket connection;
        while ((connection = listenSocket.accept()) != null) {
            RequestHandler requestHandler = new RequestHandler(connection);
            requestHandler.start();
        }
    }
}

```

# Thread 생성

```
public class RequestHandler extends Thread {
    private final static Logger log = Logger.getLogger(RequestHandler.class.getName());

    private Socket connection;

    public RequestHandler(Socket connectionSocket) {
        this.connection = connectionSocket;
    }

    public void run() {
        log.log(Level.INFO, "WebServer Thread Created!");
        InputStream is = null;
        OutputStream os = null;

        try {
            is = connection.getInputStream(); // 사용자 요청
            os = connection.getOutputStream(); // 사용자 응답

            // 구현

            connection.close();
        } catch (IOException e) {
            log.log(Level.SEVERE, e.getMessage());
        }
    }
}
```

브라우저에 응답

힌트

`OutputStream.write()` 메소드를 활용해 문자열을 반환한다.  
브라우저를 통해 확인한다.

## http header 읽기

http header example :

```
GET /?data=234 HTTP/1.1  
Host: localhost  
Connection: keep-alive  
Accept: */*
```

힌트

InputStream => InputStreamReader => BufferedReader  
BufferedReader.readLine() 메소드 활용해 header 읽는다.  
http header 전체를 출력한다.  
header 마지막 확인

```
while (!"".equals(header)) {}
```

## http method 및 url 파싱

http header example :

```
GET /?data=234 HTTP/1.1
```

힌트

StringTokenizer를 활용해 http header의 첫번째 라인 파싱  
method (GET/POST) 추출 => 첫 번째 token

url 추출 => 두 번째 token

url에서 인자 추출

```
String url = "/?data=234";  
int index = url.indexOf("?");  
String params = url.substring(index+1);
```

## url과 html 파일 매핑

Web Default Directory 설정

url

/index.html => index.html

/home.html => home.html

힌트

### Web Default Directory 설정

```
private static final String DEFAULT_WEB_ROOT = "webapps";
```

웹 서버 파일을 읽어 브라우저에 전송

```
File => FileInputStream => Socket.OuptStream.write()
```

## Response header 설정

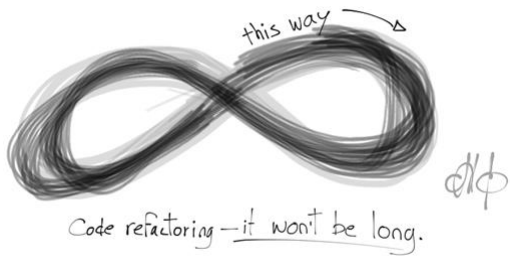
힌트

### ContentType과 Character Set 설정

```
Socket.OutputStream.writeBytes("HTTP/1.0 200 Document Follows \r\n");  
Socket.OutputStream.writeBytes("Content-Type: text/html;charset=utf-8\r\n");  
Socket.OutputStream.writeBytes("Content-Length: " + numBytes + "\r\n");
```







소스 코드 엉망이다. 리팩토링 해보자.