

컴퓨터 네트워크 Assignment 1 - WIKI

- 이름 : 문원찬
- 학번 : 2019008813

1. Design & Implement

사용한 패키지

```
// 입출력, 파일 관련 패키지
import java.io.*;
import java.nio.file.*;
import java.util.*;

// 서버, 쿠키, JSON 파싱 관련 패키지
import java.net.InetSocketAddress;
import com.sun.net.httpserver.*;
import java.net.HttpCookie;
import com.google.gson.*;
```

main 함수

```
public static void main(String[] args) throws IOException {
    HttpServer server = HttpServer.create(new InetSocketAddress(8080), 0);
    System.out.println("Listening in port: 8080");
    server.createContext("/", new RootHandler());
    server.createContext("/chair", new ChairHandler());
    server.createContext("/table", new TableHandler());
    server.createContext("/closet", new ClosetHandler());
    server.createContext("/favicon.ico", new FaviconHandler());
    server.setExecutor(null);
    server.start();
}
```

로컬호스트 8080포트에 서버를 생성하며 "Listening in port: 8080" 로그를 남긴다.
"/", "/chair", "/table", "/closet" 에 관한 컨텍스트를 설정 및 핸들러와 맵핑한다.
/favicon.ico 에 대해선 [3. Trouble Shooting]에서 설명한다.

루트 핸들러 함수

```
static class RootHandler implements HttpHandler {
    @Override
    public void handle(HttpExchange exchange) throws IOException {
        ...
    }
}
```

"Index page requested" 라는 로그를 남기며 우선 쿠키를 체크한다. 이는 따로 [쿠키 체크 함수]를 만들어 관리한다.

인덱스 HTML파일을 불러오고 byte 배열로 변환한다.

HTTP 응답 헤더에 상태코드 200(OK)와 바디의 길이를 첨부한다. 바디엔 변환한 바이트 배열을 첨부한다.

만약 파일을 불러오는데 실패했다면 404(Not Found)를 HTTP 응답 헤더에 첨부하고 "Failed : ROOT" 로그를 남긴다.

chair, table, closet 핸들러 함수

```
static class chairHandler implements HttpHandler {
    @Override
    public void handle(HttpExchange exchange) throws IOException {
        ...
    }
}
```

"[가구] page requested" 라는 로그를 남기며 쿠키를 체크한다.

세부사항 HTML파일을 불러오고 JSON으로부터 HTML을 수정한다. 이는 따로 함수로 동작한다.

수정된 HTML을 바이트 배열로 변환 후 HTTP 응답 헤더에 상태코드 200(OK)와 바이트 배열의 길이를 첨부한다. 응답 바디엔 변환한 바이트 배열을 첨부한다.

만약 파일을 불러오는데 실패했다면 404(Not Found)를 HTTP 응답 헤더에 첨부한다.

"Failed : [가구]" 로그를 남긴다.

[가구] 는 chair, table, closet 을 의미한다.

쿠키 체크 함수

```
private static HttpExchange cookieCheck(HttpExchange exchange)
```

- 입력 : HTTP 요청 및 응답 객체 (HttpExchange)
- 출력 : 수정된 HTTP 요청 및 응답 객체 (HttpExchange)
HTTP 요청 헤더에 쿠키가 있는지 체크한다. 있다면 쿠키 값을 추출하여 "Returning user, welcome [value]" 를 로그로 남긴다. 이때 [value] 는 쿠키의 값으로 본인의 학번인 "2019008813" 을 나타낼 것이다. 만약 쿠키가 없다면 "New user requested page, cookie will be set." 로그를 남긴다.
쿠키의 이름은 "Student Number", 쿠키의 값은 "2019008813" 으로 설정하고 HTTP 응답 헤더에 쿠키를 추가한다.

JSON 파싱 및 세부사항 HTML 수정 함수

```
private static String jsonToHtml(String htmlContent, int num)
```

- 입력 : HTML(string), 번호(int)
- 출력 : 수정된 HTML(string)
furniture JSON파일을 불러온다.
.com.google.gson 패키지를 사용해 JSON을 파싱한다.
입력받은 번호에 따라 [가구] 에 관련된 정보를 JSON으로부터 불러온다. (0: chair 1: table 2: closet)
입력받은 HTML에 불러온 정보를 이용해 수정한다. 이때, 이미지는 path이므로 이미지를 byte 배열로 불러와 BASE64로 인코딩해서 HTML 데이터에 삽입한다.
수정된 HTML을 리턴한다.

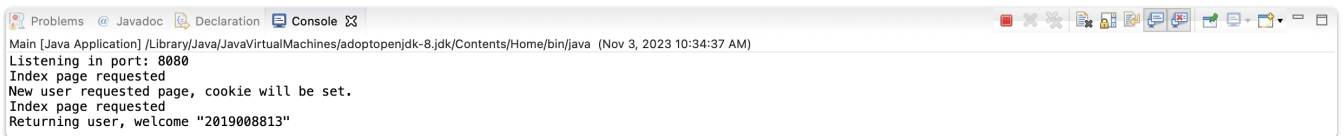
파일 읽기 함수

```
private static String readFile(String filePath)
```

- 입력 : 파일주소(string)
- 출력 : 파일(string)
HTML, JSON 파일을 줄마다 읽기 위해 사용된다.
파일의 주소를 받고 null인 줄이 나올 때까지 불러온다.
불러온 파일을 리턴한다.

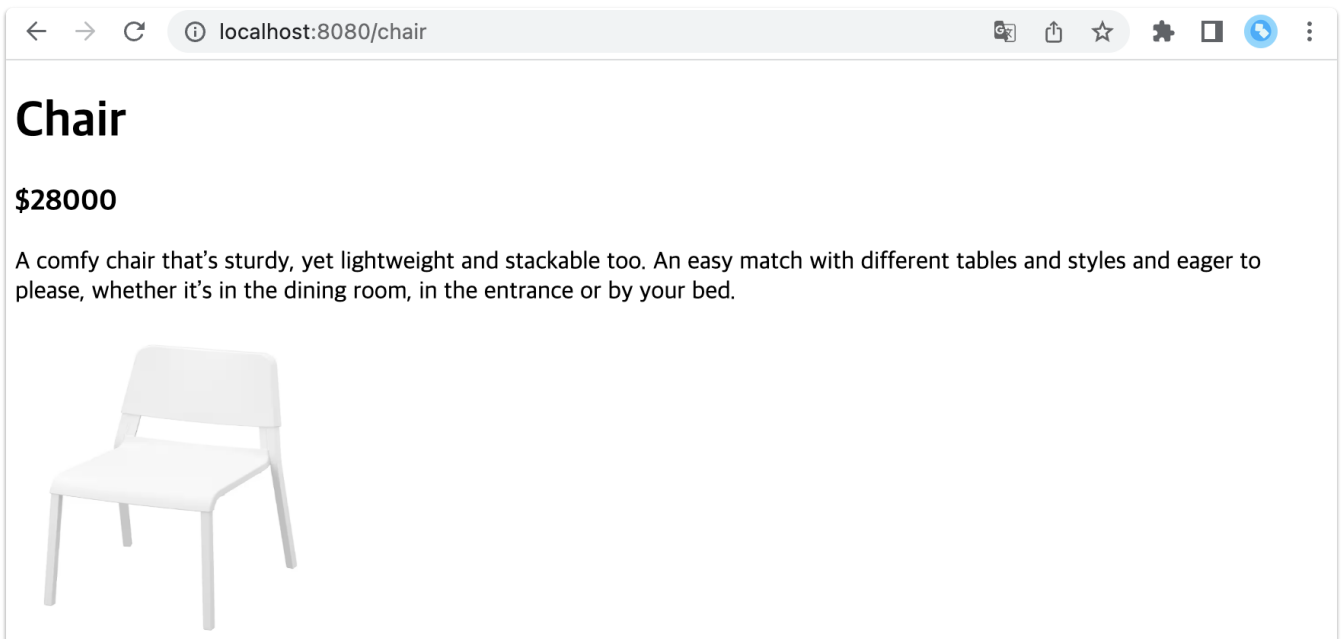
2. Result

쿠키 체크



쿠키가 없는 상태에서 서버에 접속하면 3번째 줄과 같이 새로운 유저로 인식하고 쿠키를 생성한다.
새로고침하여 다시 서버에 요청을 하면 쿠키가 존재하므로 기존의 유저로 인식한다.

세부사항 페이지



JSON 파싱을 통해 세부사항 HTML이 제대로 수정됨을 확인할 수 있다.

3. Trouble Shooting

크롬의 favicon.iso 요청

크롬 브라우저를 통해 웹서버에 접속시 기존의 요청 외에 루트주소에 대한 요청을 추가로 보내는 현상을 발견했다. 인터넷 검색을 통해 크롬은 `favicon.iso`에 대한 요청을 보내기 때문에 추가로 요청이 발생함을 알 수 있었다. 따라서 `favicon.iso`에 대한 핸들러를 따로 만들어 루트핸들러를 호출하지 못하게 함으로써 해결했다.