

웹 용어 정리

웹 개발 기본 개념

html, http, browser, url

html

- 마크업 랭귀지

html

- 데이터 전송을 위한 프로토콜
- <http 상태코드>
 - 100번대 : 대기
 - 200번대 : 통신 성공
 - 300번대 : 리다이렉션
 - 400번대 : 클라이언트 오류
 - 500번대 : 서버 오류

browser

html 문서를 읽기위한 도구

url

html 문서를 불러오기 위한 기술

Apache와 Tomcat

아파치(Apache)

- 아파치

- 웹 서버로 html, jsp, png 등의 파일을 가지고 있다.
- 서버는 기본적으로 포트를 열고 대기 상태에 있다.
- 서버는 Demon Process로 항상 가동되고 있다.

- 클라이언트

- 클라이언트가 웹 브라우저에서 아파치에게 데이터(html, jsp, png...)를 요청하는데 이때 아파치가 Java가 포함된 jsp 파일을 만나면 브라우저가 이해하지 못한다고 인식하여 jsp 파일을 톰캣에게 데이터 읽기를 위임한다.

□ 브라우저는 html, png, avi 등은 이해 가능하지만 Java는 이해 못함

톰캣(Tomcat)

Tomcat : WAS(Web Application Server)

- 아파치가 읽지 못하는 jsp 파일을 읽어 html 파일로 바꿔준다
- jsp를 java로 파싱해준다.
- 기본적으로 jsp파일이 있을 경우에만 일을 하기때문에 아파치와 톰캣을 동시에 사용함으로써 발생하는 비효율을 해결하고자 톰캣을 상시 사용하게 한다.
- uri요청(java)을 하면 무조건 톰캣이 일한다.
- b.jsp -> b.jsp.java -> (컴파일) b.jsp.class -> (실행) html코드로 변경

□톰캣이 하는 일(jsp 파일 있을 경우)

- 서블릿을 만든다.
- 컴파일한다.
- html으로 바꾼다.

MVC 모델에서의 서버 움직임

클라이언트- URI request -> 서블릿- Router(분기점, **Controller**) -> **Model**에서 데이터 가져오기 -> **View**에서 jsp찾아 응답

Stream

개울, 컴퓨터에서는 전류를 의미

- System.out : 컴퓨터 선에 연결
- System.in : 마우스, 키보드 등 연결

~Stream~

Keyboard(이하 K) -> Computer(이하 C)

K에서 흘러오는 전류는 □반도체가 받아준다.

1과 0으로 된 데이터를 인간이 이해할 수 없으므로 □부호화한다.

받은 데이터는 □램에 저장한다.

물리적으로 데이터를 주고 받으면 소멸될 가능성이 있기 때문에 예외처리를 한다. (try~catch~)

□반도체

- 열이 올라가면 전류가 흐르지 않게하고(0) 열이 떨어지면 전류가 흐르게(1) 한다. 너무 뜨거우면 반도체가 부도체가 되어버린다.

□도체

- 전류가 잘 흐르는 물질

□부도체

- 전류가 흐르지 않는 물질

□부호화 인간이 이해하기 위해 기호로 만드는 것 예) 문자(영어)

□utf-8

- 유니코드, 1~4byte의 가변길이 문자 인코딩
- 영어는 8bit(1byte)로 표현 가능하다.
- 데이터를 8bit씩 끊어 읽으면 부호화 할 수 있다.(즉 영어로 해석할 수 있다)

□RAM

- 램 저장공간 하나에 8bit = 1byte로 문자하나를 표현할 수 있다(영어)
- byte로 받으면 데이터를 부호화 시킬 수 있기 때문에 통신에서는 byte로 표현

데이터 받는 방법

1. inputStream

- 숫자로 인식하기 때문에 문자로 캐스팅해야한다.
- 1byte만 받는다.

2. inputStreamReader

- 문자로 바로 받을 수 있다.
- byte 크기를 늘릴 수는 있으나 가변길이에 대응할 수 없다.

3. BufferedReader

- 문자 가변 길이에 대응할 수 있다.

□문자 가변 길이에 대응하는 로직('hello'라는 데이터를 4byte 길이로 보낸 경우)

1. K에 데이터가 가득 차면 자동으로 flush를 해준다.
2. 일단 들어가는 만큼(hell) C에 흘려보낸다.
3. K가 비어진 곳에 나머지 문자(o)를 넣어준다.
4. K에 데이터가 꽉 차지 않아(o__) 자동으로 flush 되지 않는다.
5. readLine으로 읽으면 C에 있는 변수(data)에 담긴다(hell).
6. C에 데이터가 비워진다.
7. K를 flush해서 C로 날려준다.
8. 변수에 데이터를 쌓아두고(+=) 다시 남은 문자를 넣어준다.

PrintWriter

- BufferedReader를 내장하고 있다.
- 이를 사용하면 BufferedReader 보다 코드가 짧아진다.
- Auto flush 기능이 있다.

```
PrintWriter pw = new PrintWriter(System.out, true); // flush 안해도된다.  
pw.println("hello");
```

Stateless와 Stateful

Stateless 서버

- 연결이 지속되지 않는 서버.
- 단발적인 데이터 전달 방식으로 항상 새로운 데이터로 받아들여 이전 사용자가 누구인지 알 수 없다는 단점이 있다.
- 대표적인 예로 웹이 있다.

Stateful 서버

- 연결이 지속되는 서버.
- 지속적으로 서버에 연결되어 있어 서버에 부하가 걸린다는 단점이 있다. ex) Socket 통신
- 대표적인 예로 전화가 있다.

□Socket 통신

- 클라이언트와 서버 양쪽에서 서로에게 데이터 전달을 하는 방식의 양방향 통신.

Session과 Cookies

Session

- 전달받은 데이터를 서버에 저장하는 방식.
- Stateless 서버를 Stateful 서버인 것처럼 속이는 것.
- 클라이언트와 서버에 모두 ID를 저장.

Cookies

- 전달받은 데이터를 브라우저에 저장하는 방식
- 쿠키 확인하는 방법 : 검사 - Application - cookies

Network 탭

- 웹 브라우저 마우스 오른쪽 - 검사 - network에서 확인 가능

1. Status Code : 200

- 통신 정상적

2. Remote Address

- 내가 나를 호출

3. Response headers

- 응답할 때 헤더

4. Request headers

- 요청할 때 헤더
- GET 방식 요청 url
- 중요한 정보를 담아갈 수도 있다.

□Http의 Header와 Body

1. Header

- 클라이언트가 서버에 데이터를 요청할 때(bufferedWrite할 때) 바디 데이터를 설명해주는 역할
- 물류 공장에서 짐에 라벨을 붙이는 행위와 같은 역할

2. Body

- 요청한 데이터 부분

Request할 경우

- Get에는 body가 없고 Post의 경우에만 body가 있다.
- Get은 데이터 받기만해서 body가 필요없다.
- Post는 전화번호, 주소 등을 전송해야함으로 header 설명과 body과 필요하다.
- 설명할 때 문자를 해석하기 위해 content-Type이 꼭 필요하다.

response할 경우

- get/Post 모두 body가 있다.

Tomcat의 Scope

- Scope에는 session 영역과 request 영역이 있다.
 - 클라이언트가 서버(아파치/톰캣)에 요청을 한다.
 - 서버가 실행되면 세션 영역이 생성된다(서버 끝때까지 켜져있는 영역)
 - 모든 클라이언트는 같은 세션을 사용한다.
1. 클라이언트가 서버에 요청을 하면 랜덤으로 ID에 JSESSIONID 생성되고(예:a756) request/response(header와 body)가 만들어진다.
 2. 서버가 클라이언트에 응답을 하면 JSESSIONID를 돌려주고 연결선이 끊기며 request에 저장된 ID는 날라간다.(a756 -> 아이디 날라감) 그러나 세션에 있는 ID(a756)는 날라가지 않는다.
 3. 클라이언트가 재요청을 하면(같은 ID로 : a756) 세션에 아이디 있음을 확인하고 새로운 request/response를 생성한다.(응답할때 날라간다 -> 리퀘스트 생명주기는 요청할 때만 살아있음)

sendRedirect

- 기존 리퀘스트 사라지고 새로운 리퀘스트가 생성 (html 상태코드 300번대)
- 단점 : request를 유지할 수 없다.(모델을 뷰로 가져갈 수 없다)

Dispatcher

- 최초의 리퀘스트를 새로 만든 리퀘스트로 덮어준다. (최초의 리퀘스트임)

- 주소안바뀌고 뷰까지 끌고 갈 수 있다.
- 계속 끌고가는 거는 한계 -> 세션에 넣어 사용