


Session과 Cookie

HTTP통신방법

HTTP는 서버와 client간의 요청과 응답으로 데이터를 주고 받는 형식으로 서버는 client의 요청에 응답을 하고 나면 그 연결을 끊어버림(stateless)
client는 다시 서버에 요청하려면 새로 연결하여 응답을 받아야 한다.

문제점

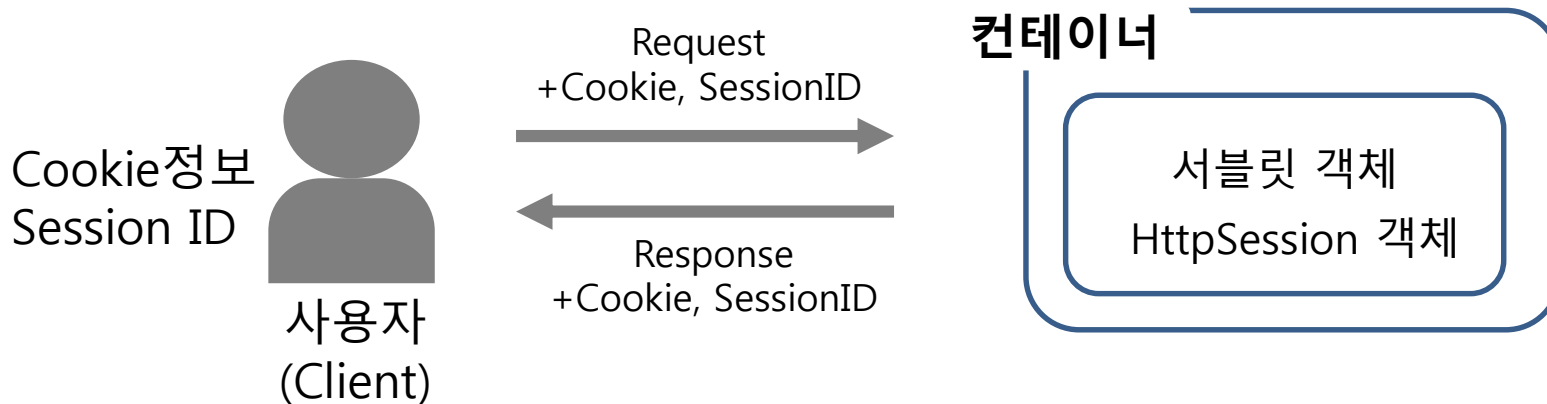
연결이 끊어지기 때문에 유지되어야 하는 정보들이 사라지는 문제가 발생을 한다. (예 로그인된 후 로그인 정보, 장바구니에 넣은 데이터 등)



Session과 Cookie

연결이 끊어진 이후에도 client에 대한 정보를 유지하기 위해 Server에서 데이터를 보관하는 방법과 client에 데이터를 보관하는 방법 두가지가 있는데 서버측에 데이터를 보관하는 방법을 Session이라고 하고, client측에서 데이터를 보관하는 방법이 Cookie이다.

** HTTP Request / Response 프로토콜 헤더에 관련 정보를 전송하여 공유함



Cookie란(패키지 : `javax.servlet.http.Cookie`)

클라이언트 즉 사용자 컴퓨터에 데이터를 저장하는 기술, 필요시에 해당하는 정보를 서버와 공유하여 정보를 유지하는 것이다.

Map형식으로 저장이 되고, 데이터의 크기, 개수에 제한이 있음,
쿠키유지시간, 유효디렉터리, 유효도메인 등을 설정할 수 있다

속성설정

name=value : ASCII문자만 사용 / 한번 설정된 쿠키의 name은 수정 못함.

expire='날짜' : 쿠키의 유지시간 설정 없으면 브라우저 동작 동안 유지

path='경로' : 쿠키가 전달되는 서버의 유효디렉터리를 지정하는 속성

domain='서버정보' : 쿠키가 전달되는 유효 서버설정

secure : https나 ssl보안프로토콜로 요청할때만 서버전송

Cookie클래스 생성(패키지 import)

```
Cookie 쿠키명 = new Cookie("명칭","값");
```

생성된 쿠키설정

setMaxAge(int expiry) : 유효시간설정

setPath(String uri) : 경로설정

** 서버의 모든 요청 X / 특정 경로를 통한 요청시 쿠키를 사용하는 경우

setDomain(String domain) : 쿠키도메인 설정, 쿠키생성

** 도메인 외의 도메인 설정시 사용

생성된 쿠키 전송

```
response.addCookie(Cookie cookie) : 생성된 쿠키전송
```

client 전송한 Cookie 읽어오기 (HttpServletRequest 객체 이용)

request.getCookies() 매소드 활용 쿠키 객체 배열로 리턴

예) `Cookie[] list=HttpRequest.getCookies();`

쿠키 값 호출

getName() / getValue()를 이용하여 이름과 값 호출

예) `Cookie[] list=HttpRequest.getCookies();`

```
for(Cookie c : list)
```

```
{
```

```
    System.out.print(c.getName()+" "+c.getValue());
```

```
}
```

** 쿠키 저장된 정보를 가지고 로그인 확인 및 팝업창 설정 등을 할 수 있음.

Explorer

브라우저 확인

개발자도구(f12) -> 네트워크 -> 페이지 선택 -> 우측화면 쿠키

저장경로

C:\Users\user1\AppData\Local\Microsoft\Windows\ INetCache

Chrome

브라우저 확인

개발자도구(f12) -> Application -> Cookies에서 확인 가능

저장경로

C:\Users\user1\AppData\Local\Google\Chrome\UserData\Default\
Cache

Session란(패키지 : `javax.servlet.http.HttpSession`)

서버에 데이터를 저장하는 기술로 client에는 Session ID를 부여하고 client가 request에 SessionID를 보내면 ID를 기준으로 일치 하는 Session정보를 컨테이너가 생성하여 그 객체의 데이터를 가져와 사용한다. 만일 client가 보낸 SessionID가 없으면 새로 객체를 생성

Session 생성

Session객체를 컨테이너가 자동으로 생성하여 request객체에 넣어주기 때문에 그 객체를 불러오는 것을 생성이라고 한다.

1. `HttpRequest.getSession();`
2. `HttpRequest.getSession(boolean);`

** SessionID일치 Session이 없을 경우

- boolean값에 따라 true : 객체생성, false : null값 반환

Session 생성(HttpServletRequest에 있음)

```
HttpSession 세션명=HttpServletRequest.getSession();
```

** client가 보낸 SessionID값이 있으면 관련 객체 호출 SessionID나 관련객체가 없으면 새로운 객체 생성

생성된 Session값 설정

```
세션명.setAttribute("이름","값(Obj)"); //세션데이터 설정
```

```
세션명.setMaxInactiveInterval(숫자); //세션유지시간설정
```

생성된 Session 호출

```
HttpSession 세션명=HttpServletRequest.getSession();
```

```
세션명.getAttribute("이름"); // 데이터불러오기
```

매소드명	내 용
setAttribute(String, object)	request객체에 전달하고 싶은 값을 String 이름으로 Object저장하는 매소드
getAttribute(String)	매개변수와 동일한 객체 속성값 불러오는매소드
getAttributeNames()	객체에 등록되어 있는 모든 정보의 이름만 반환
removeAttribute(String)	request객체에 저장되어 매개변수와 동일한 속성값 삭제하는 매소드
getId()	SessionID값을 가져오는 매소드

매소드명	내 용
getCreationTime()	Session객체가 생성된 시간을 반환(밀리초)
getMaxInactiveInterval()	클라이언트 요청이 없을때 서버가 현재의 세션을 언제까지 유지할지를 초 단위로 반환(default 30분)
getLastAccessedTime()	클라이언트 요청이 마지막으로 시도된 시간을 반환(밀리세컨초)
isNew()	새로생성된 Session이면 true 아니면 false
invalidate()	현재의 Session을 삭제
setMaxInactiveInterval(int)	객체의 유지시간을 설정, 지정된 시간이 지나면 객체 자동삭제