



# JSP 7강 – 쿠키와 세션

---

양 명 속

[[now4ever7@gmail.com](mailto:now4ever7@gmail.com)]



# 목차

---

- 쿠키
- 세션



# 쿠키

---

- 쿠키의 개요
- 쿠키의 사용
- 쿠키를 사용한 회원인증



# 쿠키의 개요

---

- 쿠키와 세션은 웹 서버와 정보의 전달을 통해 어떠한 상태를 유지할 때 사용됨
- HTTP 프로토콜의 비연결성 - 웹 브라우저의 요청에 대한 응답을 하고 나면 해당 클라이언트와의 연결을 지속하지 않음(Connectionless), stateless
- 쿠키
  - 상태가 없는(stateless) 프로토콜을 위해 상태를 지속시키기 위한 방법
  - 웹 브라우저의 정보를 웹 브라우저(클라이언트) 자신들에게 저장
  - 이후에 서버로 전송되는 요청에는 쿠키가 가지고 있는 정보가 같이 포함되어서 전송됨
  - 웹 서버는 웹 브라우저의 요청 속에 포함되어 있을 쿠키를 읽어서 새로운 웹 브라우저인지 이전에 요청을 했던 웹 브라우저인지를 판단할 수 있음
  - 웹 브라우저(클라이언트)쪽에 저장됨



# 쿠키의 개요

---

- 특정 웹 사이트에 접속할 때 생성되는 정보를 담은 임시 파일
  - 크기는 4kb 이하로 생성
  - 개인의 정보가 기록되므로 보안 문제 유발 -> 웹 브라우저 자체에 쿠키 거부 기능이 들어 있음
- 
- 세션 - 웹 서버 쪽에 저장됨
- 
- 쿠키와 세션
    - 로그인, 로그아웃, 장바구니 등의 어떤 정보의 상태 유지가 반드시 필요한 서비스를 제공해야 하는 경우에 사용



# 쿠키의 사용

---

- 쿠키 생성시 javax.servlet.http 패키지의 **Cookie** 클래스 사용
  - 생성된 쿠키 안에는 각각의 웹 브라우저를 판별할 수 있는 정보가 포함되어 있음
- (1) 쿠키 생성 및 사용
  - 쿠키의 구성요소 - **이름, 값**, 유효기간, 도메인, 경로 등
  - 쿠키의 이름 - 알파벳, 숫자로 구성
    - \$로 시작할 수 없음
    - 콤마, 세미콜론, 공백 등의 문자가 포함될 수 없음
  - 쿠키 값 - 공백, 괄호, 등호, 콤마, 콜론, 세미콜론 포함할 수 없음



# 쿠키의 사용

- 쿠키의 생성

```
Cookie cookie = new Cookie(String name, String value);
```

- name – 생성되어지는 쿠키에 대한 이름
  - value – 쿠키의 값
- 쿠키 생성 후 response 객체의 addCookie() 메서드를 사용해서 쿠키를 추가해 주어야 함

```
response.addCookie(ck);
```

- 쿠키 생성 후 쿠키의 이름에 대응하는 값을 새롭게 지정할 때 setValue() 메서드 사용

```
cookie.setValue(newValue);
```



# 쿠키의 사용

- 클라이언트의 요청과 함께 전달되어져 온 쿠키를 읽어 올 때는 request 객체의 `getCookies()` 메서드 사용

```
Cookie[] cookies = request.getCookies();
```

- `getCookies()` 메서드 - 웹 브라우저에 저장된 쿠키를 모두 읽어오기 때문에 리턴타입이 `Cookie[]`
- 쿠키의 지속시간-cookie 객체의 `setMaxAge()` 메서드 사용

```
cookie.setMaxAge(int expiry)
```

- `expiry` - 초 단위로 쿠키의 최대 수명을 설정
  - 이 시간을 벗어난 쿠키는 사용기간이 만료된 쿠키 - 더 이상 쿠키를 사용하지 않음
- 예) 쿠키 지속시간을 1주일로 설정 =>  
`cookie.setMaxAge(7*24*60*60)`



# 쿠키의 사용

## ■ (2) 쿠키 작성

### ■ 쿠키 작성 절차

- 1) 쿠키를 생성함
- 2) 쿠키에 필요한 설정을 함(예:유효시간)
- 3) 웹 브라우저에 생성된 쿠키를 전송

### ■ 웹 브라우저에 저장된 쿠키를 사용하는 절차

- 1) 웹 브라우저의 요청에서 쿠키를 얻어옴
- 2) 리턴된 쿠키의 배열에서 쿠키 이름을 가져옴
- 3) 쿠키 이름을 통해서 해당 쿠키에 설정된 값을 추출

# 쿠키를 생성하는 makeCookie.jsp

```
<%@page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<%
```

//쿠키 생성하기 - 상태 정보를 저장하기 위한 방법=> 사용자의 pc에 임시파일로 저장됨, 최대 4kb 크기

//[1]쿠키 객체 생성

Cookie cookie = new Cookie("id","hong"); //이름, 값(name, value) 쌍으로 넣는다

//[2] 유효기간 설정

cookie.setMaxAge(2\*60); //초단위로 입력 => 120초 동안 쿠키가 유지됨

//[3] 쿠키 추가

response.addCookie(cookie);

// 두번째 쿠키 생성, 추가

Cookie cookie2 = new Cookie("age","21");

cookie2.setMaxAge(20\*60); //초단위로 입력

response.addCookie(cookie2);

%>

쿠키가 생성되었습니다<br>

<br>

<a href="useCookie.jsp">쿠키 정보 읽기</a>

<br>

```
id
hong
localhost/herbmall2/cookieTest/
1024
2605856000
30295250
1405856000
30295250
*
age
21
localhost/herbmall2/cookieTest/
1024
391411968
30295267
1405856000
30295250
*
```

i.e 쿠키 폴더

C:\Users\계정

\AppData\Roaming\Microsoft\Windows\Cookies

C:\Users\계정

\AppData\Roaming\Microsoft\Windows\Cookies\Low

# 생성한 쿠키를 이용하는 useCookie.jsp

```
<%@ page language="java" contentType="text/html; charset=utf-8" pageEncoding="utf-8"%>
<%
```

```
//생성한 쿠키 정보 읽어오기
```

```
String id="", age="";
```

```
Cookie[] ckArr = request.getCookies();
```

```
if(ckArr!=null){
```

```
    for(int i=0;i<ckArr.length;i++){
```

```
        if(ckArr[i].getName().equals("id")){
```

```
            id = ckArr[i].getValue();
```

```
        }
```

```
        if(ckArr[i].getName().equals("age")){
```

```
            age = ckArr[i].getValue();
```

```
        }
```

```
    }//for
```

```
}//if
```

```
%>
```

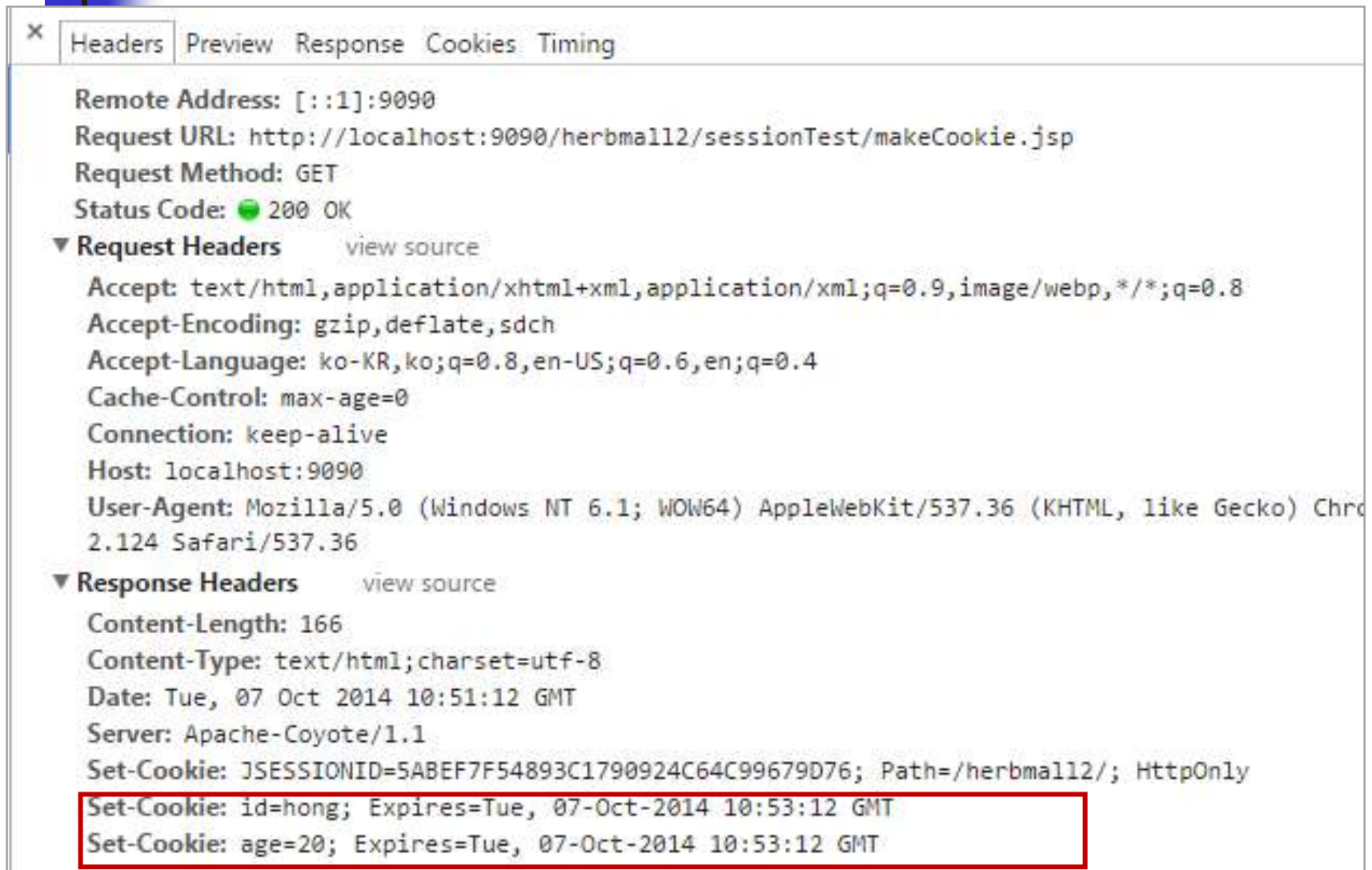
```
쿠키의 이름 : id, 쿠키의 값 : <%=id%><br>
```

```
쿠키의 이름 : age, 쿠키의 값 : <%=age%><br>
```

쿠키의 이름 : id, 쿠키의 값 : hong  
쿠키의 이름 : age, 쿠키의 값 : 21

쿠키의 유지 시간이 120초 => 120초가 지난 후에 useCookie.jsp를 재 실행하면 쿠키 값을 가져오지 못함

# makeCookie.jsp 최초 요청



Headers Preview Response Cookies Timing

Remote Address: [::1]:9090  
Request URL: http://localhost:9090/herbmall2/sessionTest/makeCookie.jsp  
Request Method: GET  
Status Code: 200 OK

▼ Request Headers [view source](#)

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8  
Accept-Encoding: gzip, deflate, sdch  
Accept-Language: ko-KR,ko;q=0.8,en-US;q=0.6,en;q=0.4  
Cache-Control: max-age=0  
Connection: keep-alive  
Host: localhost:9090  
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/2.124 Safari/537.36

▼ Response Headers [view source](#)

Content-Length: 166  
Content-Type: text/html; charset=utf-8  
Date: Tue, 07 Oct 2014 10:51:12 GMT  
Server: Apache-Coyote/1.1  
Set-Cookie: JSESSIONID=5ABEF7F54893C1790924C64C99679D76; Path=/herbmall2/; HttpOnly  
**Set-Cookie: id=hong; Expires=Tue, 07-Oct-2014 10:53:12 GMT**  
**Set-Cookie: age=20; Expires=Tue, 07-Oct-2014 10:53:12 GMT**

# useCookie.jsp 요청



X Headers Preview Response Cookies Timing

Remote Address: [::1]:9090  
Request URL: http://localhost:9090/herbmall2/sessionTest/useCookie.jsp  
Request Method: GET  
Status Code: 200 OK

▼ Request Headers [view source](#)

- Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8
- Accept-Encoding: gzip,deflate,sdch
- Accept-Language: ko-KR,ko;q=0.8,en-US;q=0.6,en;q=0.4
- Connection: keep-alive
- Cookie: id=hong; age=20; JSESSIONID=5ABEF7F54893C1790924C64C99679D76**
- Host: localhost:9090
- Referer: http://localhost:9090/herbmall2/sessionTest/makeCookie.jsp
- User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) 2.124 Safari/537.36

▼ Response Headers [view source](#)

- Content-Length: 81
- Content-Type: text/html; charset=utf-8
- Date: Tue, 07 Oct 2014 10:54:21 GMT
- Server: Apache-Coyote/1.1



# 쿠키

---

- 쿠키 - 도메인 지정
  - `Cookie.setDomain()`
  - 도메인 인자 형식
    - `.herbmall.com` : 점으로 시작하는 경우 관련 도메인에 모두 쿠키를 전송한다.
      - 예) `mail.herbmall.com`, `www.herbmall.com`, `javacan.herbmall.com` 으로 모두 전송
    - `www.herbmall.com` : 특정 도메인에 대해서만 쿠키를 전송한다.
    - 별도로 도메인을 설정 하지 않으면 쿠키를 생성한 서버에만 전달된다.



# 쿠키

---

- 쿠키 - 경로 지정
  - 쿠키에서 사용하는 경로는 디렉터리 수준의 경로를 사용한다.
  - `Cookie.setPath()`
    - 경로 미 설정 : 요청 URL의 경로에 대해서만 쿠키 전달
    - 경로 설정 : 설정한 경로 및 그 하위 경로에 대해서 쿠키 전달

```
ex)
Cookie cookie = new Cookie("name", "value");
cookie.setPath("/test");
=> name 쿠키를 /test 또는 그 하위 경로에만 전송한다.
```

- 일반적으로 쿠키는 웹 어플리케이션에 포함된 다수의 JSP 및 서블릿에서 공통으로 사용되는 경우가 많기 때문에, 대부분 쿠키의 경로를 "/"으로 지정한다.



# 쿠키

---

- [1] path 지정 안 한 경우

- 이름: ck\_userid
- 콘텐츠: lee
- 도메인: localhost
- 경로: /herbmall/login
- 다음에 대해 보내기: 모든 종류의 연결
- 스크립트에 액세스 가능: 예
- 생성됨: 2016년 11월 15일 화요일 오전 11:01:23
- 만료되는 쿠키: 2019년 8월 12일 월요일 오전 11:01:23

- [2] path 지정한 경우

- 이름: ck\_userid
- 콘텐츠: 44
- 도메인: localhost
- 경로: /
- 다음에 대해 보내기: 모든 종류의 연결
- 스크립트에 액세스 가능: 예
- 생성됨: 2016년 11월 15일 화요일 오후 2:13:10
- 만료되는 쿠키: 2019년 8월 12일 월요일 오후 2:13:10





# 쿠키-한글 처리

---

- 쿠키 이름 및 값에 대한 규칙(RFC2109규약)
  - [1] 쿠키 이름은 아스키코드 알파벳과 숫자만 포함할 수 있다.
  - [2] 콤마(,), 세미콜론(;), 공백(' ') 등의 문자는 포함할 수 없다.
  - [3] \$로 시작할 수 없다.
- => 위와 같은 문자들이 포함되도록 하기 위해서는 반드시 인코딩 처리를 해줘야 함
  - ex) `Cookie c = new Cookie("name", URLEncoder.encode("홍길동"));`
  - 쿠키 값을 읽을 때는 `URLDecoder.decode(cks[i].getValue());`



# 세션

---



# 세션

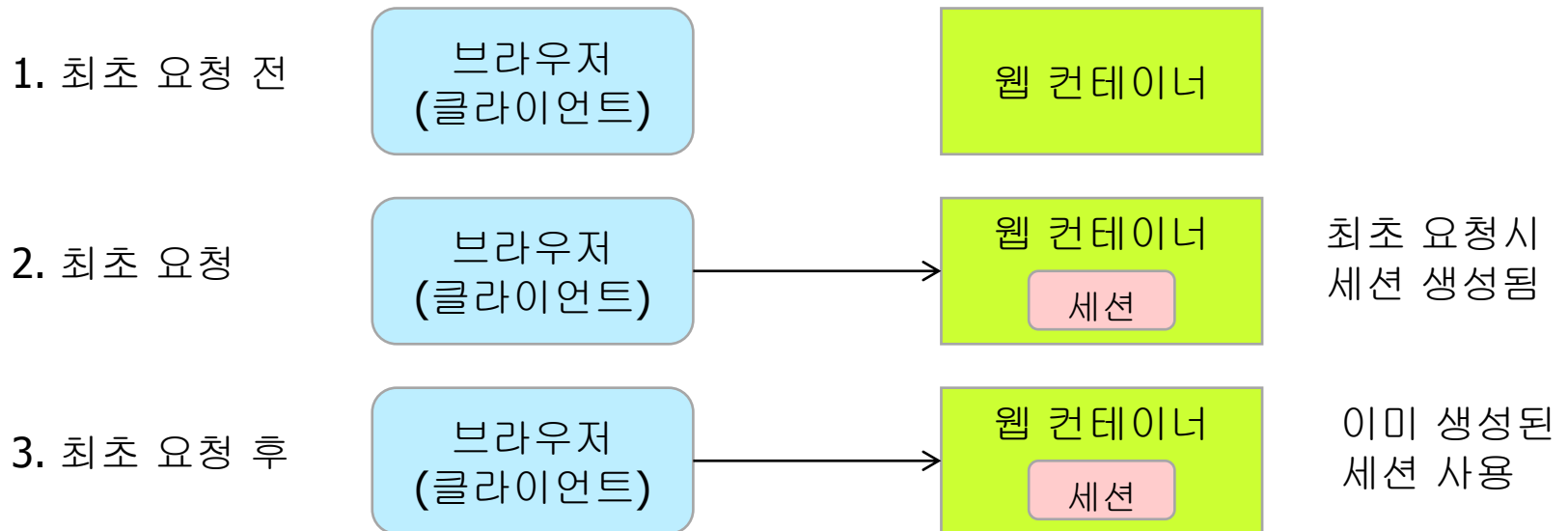
---

## ■ 세션의 개요

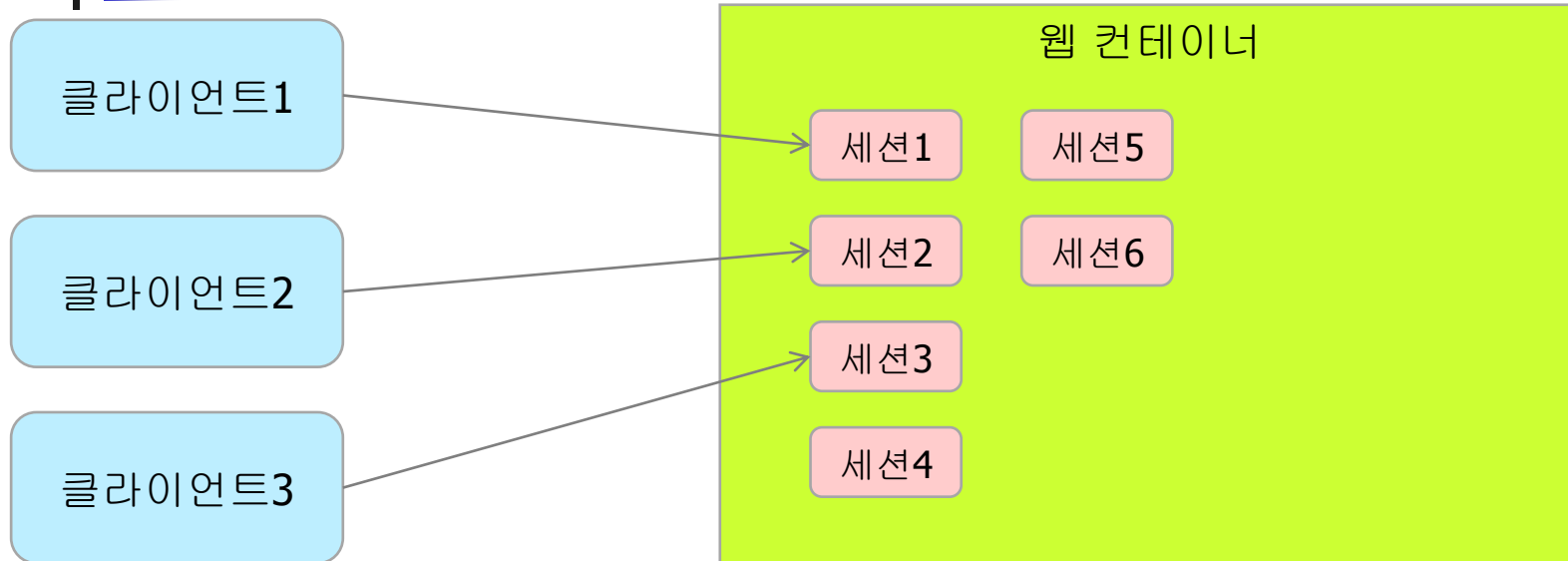
- 웹 서버 쪽의 웹 컨테이너에 상태를 유지하기 위한 정보를 저장
- cf. 쿠키 - 웹 브라우저(클라이언트)쪽에 상태를 유지하기 위한 정보를 저장
  - 웹 브라우저에 저장되어 웹 서버가 쿠키 정보를 읽어서 사용
  - 보안상 문제
- 세션 - javax.servlet.http 패키지의 HttpSession 인터페이스를 구현해서 사용
- 세션은 웹 브라우저(클라이언트)당 1개씩 생성되어 웹 컨테이너에 저장됨

# 세션

- 세션은 웹 브라우저가 최초로 접속할 때 생성되며, 그 후로는 이미 생성된 세션이 사용됨



# 세션





## 세션

---

- 클라이언트(웹 브라우저)마다 별도의 세션을 갖게 되는데, 이 때 각각의 세션을 구분하기 위해서 세션마다 고유의 ID를 할당하며, 그 아이디를 세션 ID(JSESSIONID) 라고 함
- 웹 컨테이너는 웹 브라우저에 세션 ID를 전송하며, 웹 브라우저는 웹 컨테이너에 연결할 때마다 매번 세션 ID를 알려주어서 어떤 세션을 사용할 지 판단할 수 있게 해줌
- 세션 ID는 쿠키를 통해서 웹 브라우저에 전달됨




# 세션

---

- 세션과 웹 브라우저의 관계
  - 웹 서버는 각각의 웹 브라우저로부터 발생한 요청에 대해서 특정한 식별자를 부여함
  - 이 후에 이 식별자를 웹 브라우저에서 발생한 요청들과 비교해서 같은 식별자인지 구별
  - 이 특별한 식별자에 특정한 값을 넣을 수 있기 때문에 이것을 사용해서 세션을 유지하게 됨

# 세션

<http://localhost:9090/herbmall2/member/agreement.jsp> 최초 접속



Headers Preview Response Cookies Timing

Remote Address: [::1]:9090  
Request URL: http://localhost:9090/herbmall2/member/agreement.jsp  
Request Method: GET  
Status Code: 200 OK

▼ Request Headers view source

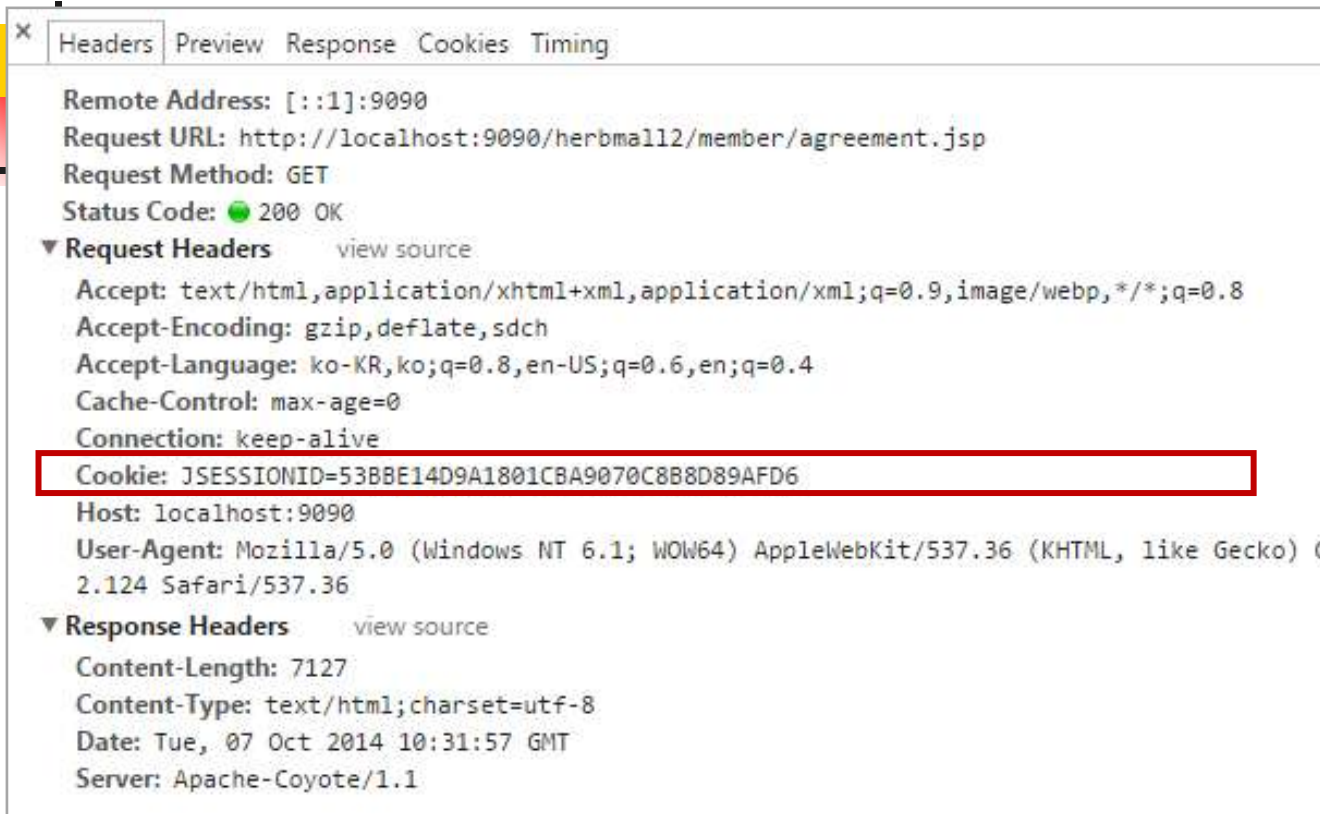
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,\*/\*;q=0.8  
Accept-Encoding: gzip,deflate,sdch  
Accept-Language: ko-KR,ko;q=0.8,en-US;q=0.6,en;q=0.4  
Cache-Control: max-age=0  
Connection: keep-alive  
Host: localhost:9090  
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/2.124 Safari/537.36

▼ Response Headers view source

Content-Length: 7127  
Content-Type: text/html; charset=utf-8  
Date: Tue, 07 Oct 2014 10:24:00 GMT  
Server: Apache-Coyote/1.1  
Set-Cookie: JSESSIONID=538BE14D9A1801CBA9070C8B8D89AFD6; Path=/herbmall2/; HttpOnly

- Response Headers에 쿠키 정보를 client로 전달한 것을 볼 수 있다. JSESSIONID 값이 톰캣에서 발급하는 쿠키 정보이다.
- 톰캣은 최초 접속 시 JSESSIONID 쿠키 값을 브라우저에 내려준다.

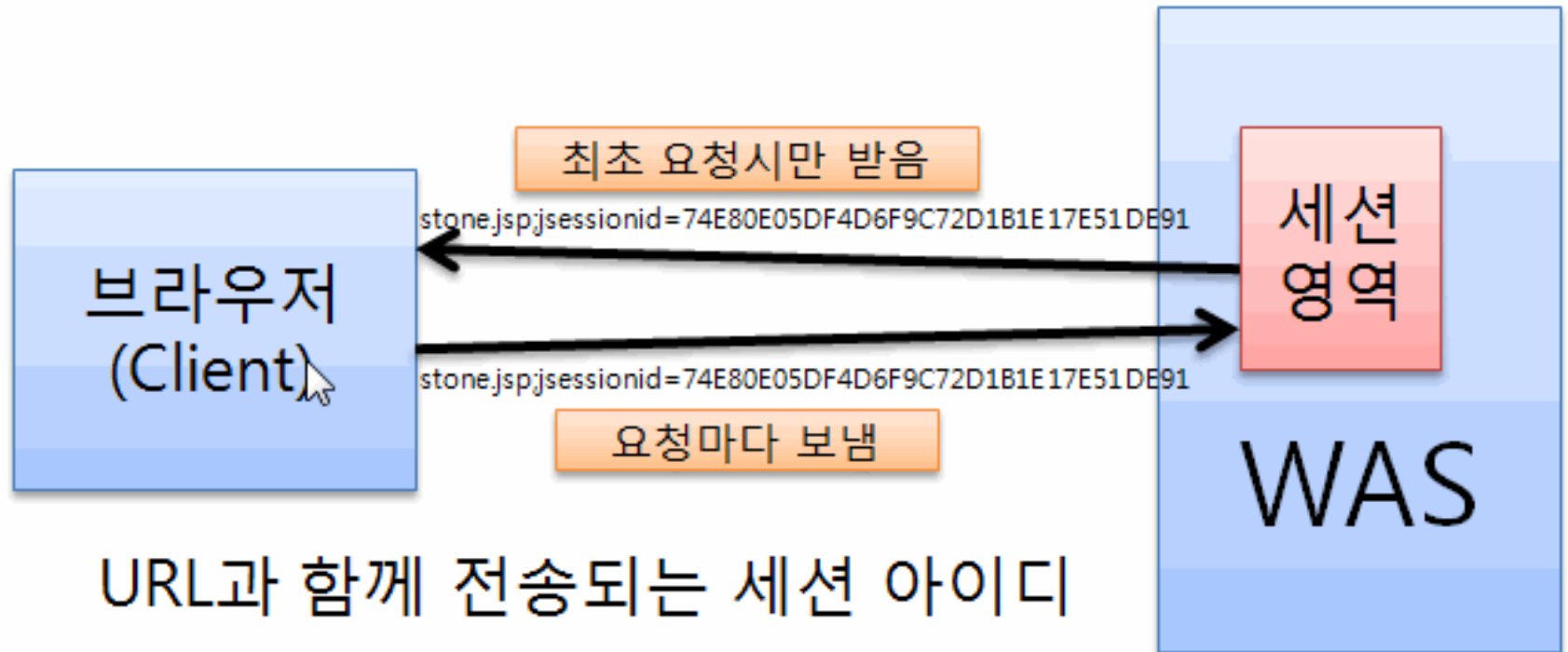




- 재요청을 보내게 되면 **client** 브라우저는 로컬에 저장되어 있는 **JSESSIONID** 쿠키 값을 **request header**에 담아 서버로 전송한다.
- 톱캣은 **client**로부터 **JSESSIONID** 쿠키 값을 전달 받으면 **response**로 **JSESSIONID** 쿠키 값을 내려주지 않는다.

- 로그인 페이지에서 사용자가 로그인 정보를 입력 후 서버로 전송을 하면, 웹 애플리케이션에서 사용자 인증 후, 세션 객체에 **JSESSIONID** 쿠키 값을 키로 사용하는 사용자 정보를 메모리 영역에 저장하게 된다.
- 키 값으로 사용되는 세션 아이디 값이 메모리에 저장 됨으로써 사용자는 로그인 상태를 유지하게 됨

# 세션





# 쿠키 대신 세션을 사용하는 이유

- [1] 세션이 쿠키보다 보안에서 앞서기 때문
  - 쿠키의 이름이나 데이터는 네트워크를 따라서 전달되기 때문에 보안에 취약
  - 세션의 값은 오직 서버에만 저장되기 때문에 중요한 데이터를 저장하기에 알맞은 장소임
- [2] 웹 브라우저가 쿠키를 지원하지 않을 경우 또는 강제로 쿠키를 막은 경우 쿠키는 사용할 수 없게 되지만, 세션은 쿠키 설정 여부에 상관없이 사용할 수 있으므로

- session은 내부적으로 cookie를 사용하거나 URL Rewriting을 사용함
- 쿠키를 사용하지 못하도록 설정되었을 경우 URL Rewriting을 사용
  - URL Rewriting은 각 URL의 끝 부분에 세션을 식별할 수 있는 추가적인 정보(jsessionid)를 붙인다.

예) <http://aa/b.jsp;jsessionid=03BA4F680646E2250ECA9DEAD0589073>



# 세션의 사용

---

- [1] 세션 속성
- Session 객체의 메소드들
  - Object **getAttribute**(String name)
  - Enumeration **getAttributeNames**()
  - long **getCreationTime**()
  - String **getId**()
  - int **getMaxInactiveInterval**()
  - void **invalidate**() – 현재 생성된 세션을 무효화시킴
  - void **removeAttribute**(String name)
  - void **setAttribute**(String name, Object value)
  - void **setMaxInactiveInterval**(int interval)



# 세션의 유효시간 지정

- [1] session 기본 객체가 제공하는 `setMaxInactiveInterval()` 메서드 사용

- 예) 세션 유효시간을 50분으로 지정

```
session.setMaxInactiveInterval(50*60); //초 단위
```

- [2] web.xml 파일에서 지정

- `<session-timeout>` 태그의 값이 세션의 유효시간이 되는데, 이 값의 단위는 분이다

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee" ...>

    <session-config>
        <session-timeout>50</session-timeout>
    </session-config>

</web-app>
```



# 세션 속성

- 세션 속성의 설정

```
session.setAttribute("userid", "hong"); //속성이름, 속성값
```

- 세션의 속성 사용

```
String id = (String)session.getAttribute("userid");
```

- 주어진 속성명에 해당하는 속성값을 얻어올 때 사용

- 세션의 속성 삭제

```
session.removeAttribute("userid");
```

- 세션의 모든 속성 삭제

```
session.invalidate();
```

## 예제

http://localhost:9090/projectStudy2/cookie/setSession.jsp

세션에 id 와 pwd 속성 설정 완료!!

세션 속성확인

http://localhost:9090/projectStudy2/cookie/viewSession.jsp

세션의 속성명 : id , 세션의 속성값 : hong  
세션의 속성명 : pwd , 세션의 속성값 : 1234



# 예제-session.jsp

세션의 속성 설정

```
<%@ page language="java" contentType="text/html; charset=utf-8"
    pageEncoding="utf-8"%>
<%
//Session => 사용자의 정보를 저장하는 방법 => 웹 서버에 저장됨
String id = "hong";
String pwd="1234";

session.setAttribute("id", id); //name, value 쌍으로 넣는다
session.setAttribute("pwd", pwd);
%>
session에 정보가 저장되었습니다.
<br>
<a href="useSession.jsp">세션 정보 읽어오기</a>
<br>
session id : <%=session.getId() %>
```





# useSession.jsp

세션 속성의 이름과 값 확인

```
<%@ page language="java" contentType="text/html; charset=utf-8"
    pageEncoding="utf-8"%>
<%
String id = (String)session.getAttribute("id");
String pwd = (String)session.getAttribute("pwd");
int interval = session.getMaxInactiveInterval(); //기본-30분(1800초)
%>
세션에서 읽은 아이디 : <%=id%><br>
비밀번호 : <%=pwd%>
<br>
세션 유지 시간 : <%=interval %>초
<br>
session id : <%=session.getId() %>
<br>
<a href="logout.jsp" >로그아웃</a>
```



# logout.jsp

---

```
<%
```

```
    //세션의 모든 정보 삭제
```

```
    session.invalidate();
```

```
%>
```

```
세션 아이디:<%=session.getId() %><br><br>
```

```
<a href="useSession.jsp">useSession 페이지로 이동</a>
```