

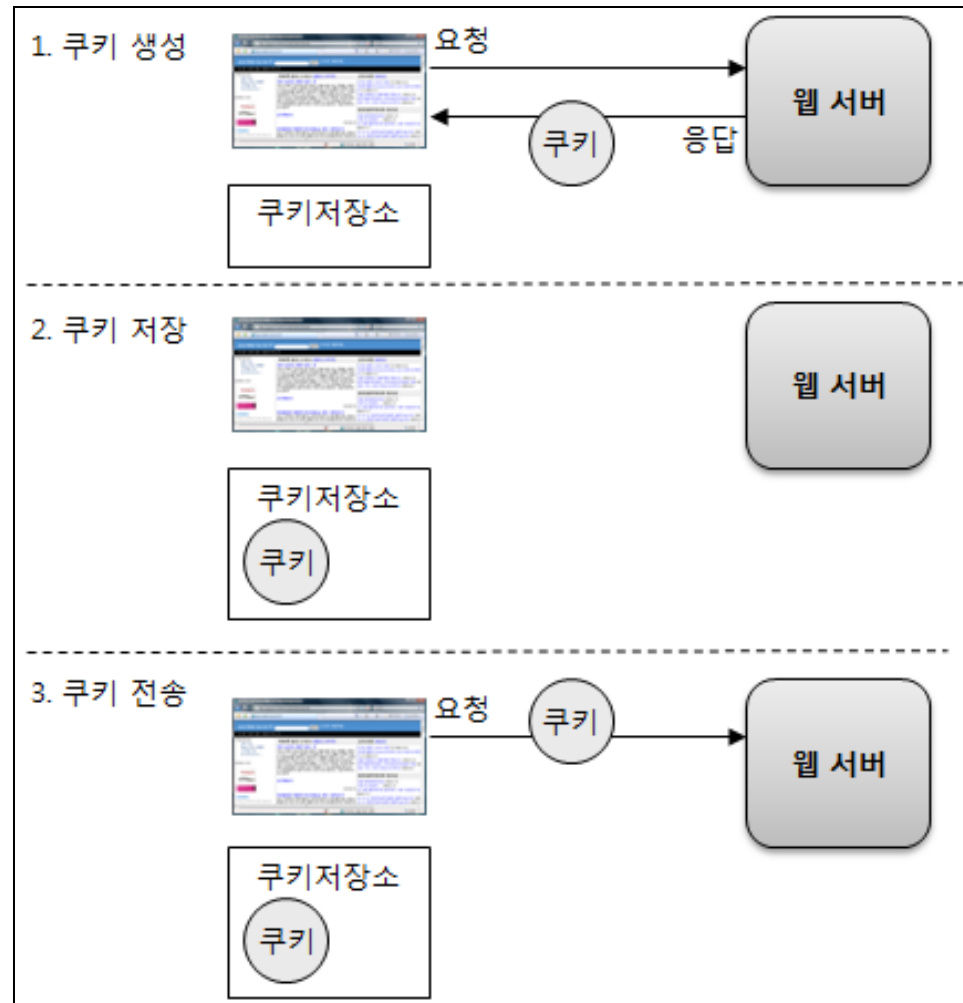
# JSP ( Java Sever Page )

- 클라이언트와의 대화 - 쿠키

- 쿠키 사용
  - 쿠키 구성
  - 생성
  - 값 사용
  - 도메인/경로/유효시간
  - 쿠키와 헤더

## ■ 쿠키(Cookie)

- '쿠키(cookie)'는 웹 브라우저가 보관하고 있는 데이터로서 웹 서버에 요청을 보낼 때 함께 전송
- 동작 방식



## ■ 쿠키의 구성

- 구성 요소
  - 이름 : 각각의 쿠키를 구별하는 데 사용되는 이름
  - 값 : 쿠키의 이름과 관련된 값
  - 유효시간 : 쿠키의 유지 시간
  - 도메인 : 쿠키를 전송할 도메인
  - 경로 : 쿠키를 전송할 요청 경로
- 쿠키 이름의 제약
  - 쿠키의 이름은 아스키 코드의 알파벳과 숫자만을 포함할 수 있다.
  - 콤마(,), 세미콜론(;), 공백(' ') 등의 문자는 포함할 수 없다.
  - '\$'로 시작할 수 없다.

## ■ JSP에서 쿠키 생성 / 읽기

- Cookie 클래스를 이용해서 쿠키 생성

```
<%  
    Cookie cookie = new Cookie("cookieName", "cookieValue");  
    response.addCookie(cookie);  
%>
```

- 클라이언트가 보낸 쿠키 읽기

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

- 읽기 관련 주요 메서드

메서드	리턴타입	설명
getName()	String	쿠키의 이름을 구한다.
getValue()	String	쿠키의 값을 구한다.

## ■ JSP에서 쿠키 생성 / 읽기

### WmakeCookie.jsp

```
<%@ page contentType = "text/html; charset=utf-8" %>
<%@ page import = "java.net.URLEncoder" %>
<%
Cookie cookie = new Cookie("name", URLEncoder.encode("유영진"));
response.addCookie(cookie);
%>
<html>
<head> <title> 쿠키 생성 </title> </head>
<body>

<%= cookie.getName() %> 쿠키의 값 = "<%= cookie.getValue() %>"

</body>
</html>
```

## ■ 쿠키 값의 인코딩/디코딩 처리

- 쿠키는 값으로 한글과 같은 문자를 가질 수 없음
  - 쿠키의 값을 인코딩해서 지정할 필요 있음
- 쿠키 값의 처리
  - 값 설정시 : `URLEncoder.encode("값", "utf-8")`
    - 예, `new Cookie("name", URLEncoder.encode("값", "utf-8"));`
  - 값 조회시 : `URLDecoder.decode("값", "utf-8")`
    - `Cookie cookie = ...;`  
`String value = URLDecoder.decode(cookie.getValue(), "utf-8");`

# ■ 쿠키 값의 인코딩/디코딩 처리

## WmakeCookie.jsp

```
<%@ page contentType = "text/html; charset=utf-8" %>
<%@ page import = "java.net.URLDecoder" %>
<html>
<head><title>쿠키 목록</title> </head>
<body>
쿠키 목록<br>
<%
    Cookie[] cookies = request.getCookies();
    if (cookies != null && cookies.length > 0) {
        for (int i = 0 ; i < cookies.length ; i++) {
            <%= cookies[i].getName() %> =
            <%= URLDecoder.decode(cookies[i].getValue(), "utf-8") %> <br>
        }
    } else { %>
쿠키가 존재하지 않습니다.
<% } %>
</body>
</html>
```



## ■ 쿠키 값 변경

- 기존에 존재하는 지 확인 후, 쿠키 값 새로 설정

```
Cookie[] cookies = request.getCookies();
if (cookies != null && cookies.length > 0) {
    for (int i = 0 ; i < cookies.length ; i++) {
        if (cookies[i].getName().equals("name")) {
            Cookie cookie = new Cookie(name, value);
            response.addCookie(cookie);
        }
    }
}
```

## ■ 쿠키 값 변경

### WmodifyCookie.jsp

```
<%@ page contentType = "text/html; charset=utf-8" %>
<%@ page import = "java.net.URLEncoder" %>
<%
    Cookie[] cookies = request.getCookies();
    if (cookies != null && cookies.length > 0) {
        for (int i = 0 ; i < cookies.length ; i++) {
            if (cookies[i].getName().equals("name")) {
                Cookie cookie = new Cookie("name",
                    URLEncoder.encode("JSP프로그래밍", "utf-8"));
                response.addCookie(cookie);
            }
        }
    }
%>
<html>
<head><title>값 변경</title></head>
<body>
name 쿠키의 값을 변경합니다.
</body>
</html>
```

## ■ 쿠키 값 삭제

- 기존에 존재하는 지 확인 후, 쿠키 값 새로 설정

```
Cookie cookie = new Cookie(name, value);  
cookie.setMaxAge(0); //유효시간 = 0  
response.addCookie(cookie);
```

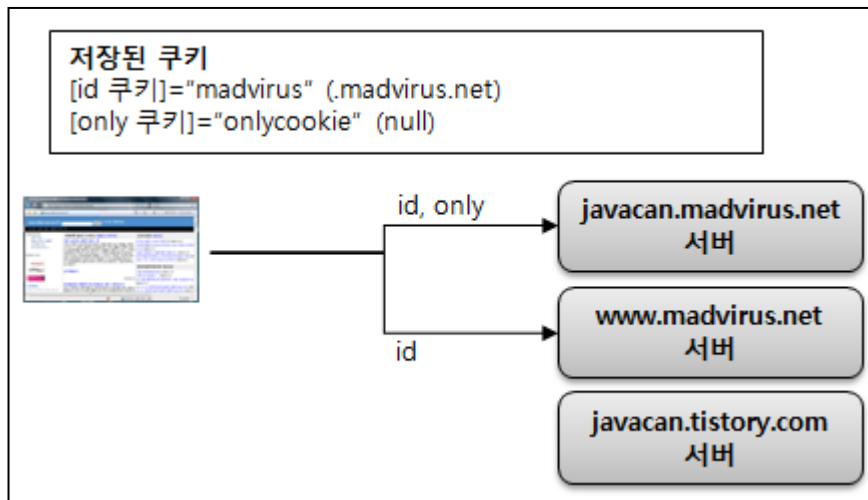
## ■ 쿠키 값 삭제

### WdeleteCookie.jsp

```
<%@ page contentType = "text/html; charset=utf-8" %>
<%@ page import = "java.net.URLEncoder" %>
<%
    Cookie[] cookies = request.getCookies();
    if (cookies != null && cookies.length > 0) {
        for (int i = 0 ; i < cookies.length ; i++) {
            if (cookies[i].getName().equals("name")) {
                Cookie cookie = new Cookie("name", "");
                cookie.setMaxAge(0);
                response.addCookie(cookie);
            }
        }
    }
%>
<html>
<head><title>값 변경</title></head>
<body>
name 쿠키의 값을 변경합니다.
</body>
</html>
```

## ■ 쿠키의 도메인과 경로

- 도메인 지정 시, 해당 도메인에 쿠키 전달
  - `Cookie.setDomain()`으로 쿠키 설정
  - 도메인 형식
    - `.madvirus.net` : 점으로 시작하는 경우 관련 도메인에 모두 쿠키를 전송한다.
    - `www.madvirus.net` - 특정 도메인에 대해서만 쿠키를 전송한다.
- 웹 브라우저는 도메인이 벗어난 쿠키는 저장하지 않음
- 쿠키 도메인에 따라 쿠키가 전달



only 쿠키를 javacan.madvirus.net 서버에서 생성했다고 한 경우

## ■ 쿠키의 도메인과 경로

### WmakeCookieWithDomain.jsp

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

```
<%@ page import = "java.net.URLEncoder" %>
```

```
<%
```

```
    Cookie cookie1 = new Cookie("id", "madvirus");
```

```
    cookie1.setDomain(".somehost.com");
```

```
    response.addCookie(cookie1);
```

```
    Cookie cookie2 = new Cookie("only", "onlycookie");
```

```
    response.addCookie(cookie2);
```

```
    Cookie cookie3 = new Cookie("invalid", "invalidcookie");
```

```
    cookie3.setDomain("javacan.tistory.com");
```

```
    response.addCookie(cookie3);
```

```
%>
```

## ■ 쿠키의 도메인과 경로

### WmakeCookieWithDomain.jsp

```
<html>  
<head> <title> 쿠키 생성 </title> </head>  
<body>
```

다음과 같이 쿠키를 생성했습니다.<br>

```
<%= cookie1.getName() %>=<%= cookie1.getValue() %>  
[<%= cookie1.getDomain() %>]  
<br>  
<%= cookie2.getName() %>=<%= cookie2.getValue() %>  
[<%= cookie2.getDomain() %>]  
<br>  
<%= cookie3.getName() %>=<%= cookie3.getValue() %>  
[<%= cookie3.getDomain() %>]  
  
</body>  
</html>
```

## ■ 쿠키의 유효 시간

- 유효 시간
  - 유효 시간 미 지정시, 웹 브라우저 닫을 때 쿠키도 함께 삭제
  - `Cookie.setMaxAge()`로 쿠키 유효 시간 설정
    - 유효 시간이 지나지 않을 경우 웹 브라우저를 닫더라도 쿠키가 삭제되지 않고, 이후 웹 브라우저를 열었을 때 해당 쿠키 전송됨
    - 유효 시간 : 초 단위로 설정



## ■ 쿠키의 유효 시간

### makeCookieWithMaxAge.jsp

```
<%@ page contentType = "text/html; charset=utf-8" %>
<%
    Cookie cookie = new Cookie("oneh", "1time");
    cookie.setMaxAge(60 * 60); // 60초(1분) * 60 = 1시간
    response.addCookie(cookie);
%>
<html>
<head> <title>쿠키유효시간설정 </title> </head>
<body>

유효시간이 1시간인 oneh 쿠키 생성.

</body>
</html>
```

## ■ 쿠키처리를 위한 유틸리티 클래스

### src/util/CookBox.java

```
package util;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.Cookie;
```

```
import java.util.Map;
```

```
import java.net.URLEncoder;
```

```
import java.net.URLDecoder;
```

```
import java.io.IOException;
```

```
public class CookieBox {
```

```
    private Map<String, Cookie> cookieMap = new java.util.HashMap<String, Cookie>();
```

```
    public CookieBox(HttpServletRequest request) {
```

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

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

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

```
                cookieMap.put(cookies[i].getName(), cookies[i]);
```

```
            }
```

```
        }
```

## ■ 쿠키처리를 위한 유틸리티 클래스

src/util/CookBox.java

```
}  
public static Cookie createCookie(String name, String value)  
throws IOException {  
    return new Cookie(name, URLEncoder.encode(value, "utf-8"));  
}  
public static Cookie createCookie(String name, String value, String path,  
    int maxAge) throws IOException {  
    Cookie cookie = new Cookie(name, URLEncoder.encode(value, "utf-8"));  
    cookie.setPath(path);  
    cookie.setMaxAge(maxAge);  
    return cookie;  
}  
public static Cookie createCookie(String name, String value, String domain,  
    String path, int maxAge) throws IOException {  
    Cookie cookie = new Cookie(name, URLEncoder.encode(value, "utf-8"));  
    cookie.setDomain(domain);  
    cookie.setPath(path);  
    cookie.setMaxAge(maxAge);  
    return cookie;
```

## ■ 쿠키처리를 위한 유틸리티 클래스

src/util/CookBox.java

```
}

public Cookie getCookie(String name) {
    return cookieMap.get(name);
}

public String getValue(String name) throws IOException {
    Cookie cookie = cookieMap.get(name);
    if (cookie == null) {
        return null;
    }
    return URLDecoder.decode(cookie.getValue(), "utf-8");
}

public boolean exists(String name) {
    return cookieMap.get(name) != null;
}
}
```

## ■ 쿠키처리를 위한 유틸리티 클래스

### WmakeCookieUsingBox.jsp

```
<%@ page contentType = "text/html; charset=utf-8" %>
<%@ page import = "util.CookieBox" %>
<%
    response.addCookie(CookieBox.createCookie("name", "홍길동"));
    response.addCookie(
        CookieBox.createCookie("id", "gdhong", "/chap09", -1));
%>
<html>
<head> <title>CookieBox사용예 </title> </head>
<body>

CookieBox를 사용하여 쿠키 생성

</body>
</html>
```

## ■ 쿠키처리를 위한 유틸리티 클래스

### WreadCookieUsingBox.jsp

```
<%@ page contentType = "text/html; charset=utf-8" %>
<%@ page import = "util.CookieBox" %>
<%
    CookieBox cookieBox = new CookieBox(request);
%>
<html>
<head><title>Cookie 사용</title></head>
<body>

name 쿠키 = <%= cookieBox.getValue("name") %> <br>
<% if (cookieBox.exists("id")) { %>
id 쿠키 = <%= cookieBox.getValue("id") %> <br>
<% } %>
</body>
</html>
```

- 추가 구현 요청 사항

ID 저장하기 체크박스 구현하시오.

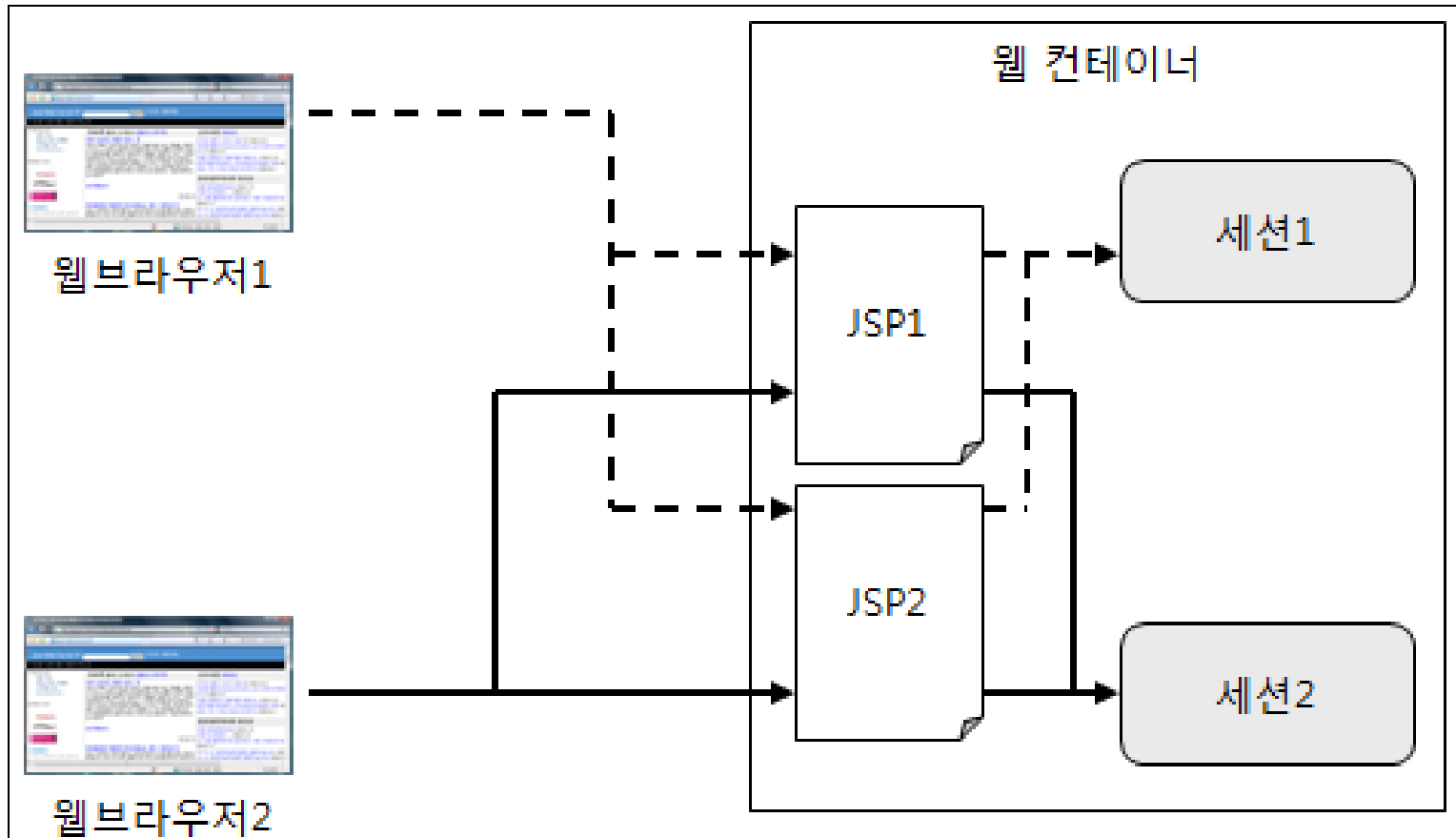
## ■ CONTENTS

- 세션: session 기본 객체
- 세션 종료 / 유효 시간



## ■ 세션(session)이란

- 웹 컨테이너에서 클라이언트의 정보를 보관할 때 사용
- 오직 서버에서만 생성
- 클라이언트마다 세션이 생성



## ■ 세션과 session 기본 객체

- page 디렉티브의 session 속성 값을 true로 지정
  - 세션이 존재하지 않을 경우 세션이 생성되고, 세션이 존재할 경우 이미 생성된 세션을 사용
- session 기본 객체를 이용해서 세션에 접근
  - session의 기본 값은 true이므로 false로 하지 않는 이상 항상 세션 사

용

```
<%@ page contentType = ... %>
<%@ page session = "true" %>
<%
    ...
    session.setAttribute("userInfo", userInfo);
    ...
%>
```

- 속성 이용해서 클라이언트 관련 정보 저장

## ■ 세션과 session 기본 객체

- session 정보 관련 메서드

메서드	리턴타입	설명
getId()	String	세션고유 ID를 구한다. (세션 ID라고 한다.)
getCreationTime()	long	세션이 생성된 시간을 구한다. 시간은 1970년 1월 1일 이후 흘러간 시간을 의미하며, 단위는 1/1000초 이다.
getLastAccessedTime()	long	웹 브라우저가 가장 마지막에 세션에 접근한 시간을 구한다. 시간은 1970년 1월 1일 이후 흘러간 시간을 의미하며, 단위는 1/1000초 이다.

## ■ 세션과 session 기본 객체

### WsessionInfo.jsp

```
<%@ page contentType = "text/html; charset=euc-kr" %>
<%@ page session = "true" %>
<%@ page import = "java.util.Date" %>
<%@ page import = "java.text.SimpleDateFormat" %>
<%
    Date time = new Date();
    SimpleDateFormat formatter =
        new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
%>
<html>
<head><title>세션정보</title></head>
<body>
세션ID: <%= session.getId() %> <br>
<%      time.setTime(session.getCreationTime());      %>
세션생성시간: <%= formatter.format(time) %> <br>
<%      time.setTime(session.getLastAccessedTime());  %>
최근접근시간: <%= formatter.format(time) %>
</body>
</html>
```

## ■ 세션과 session 기본 객체

### WsetMemberInfo.jsp

```
<%@ page contentType = "text/html; charset=euc-kr" %>
```

```
<%  
  
    session.setAttribute("MEMBERID", "madvirus");  
    session.setAttribute("NAME", "최범균");
```

```
%>  
  
<html>  
<head> <title> 세션에 정보 저장 </title> </head>  
<body>
```

세션에 정보를 저장하였습니다.

```
<%      String name = (String)session.getAttribute("NAME");           %>  
회원명 : <%= name %>
```

```
</body>  
</html>
```

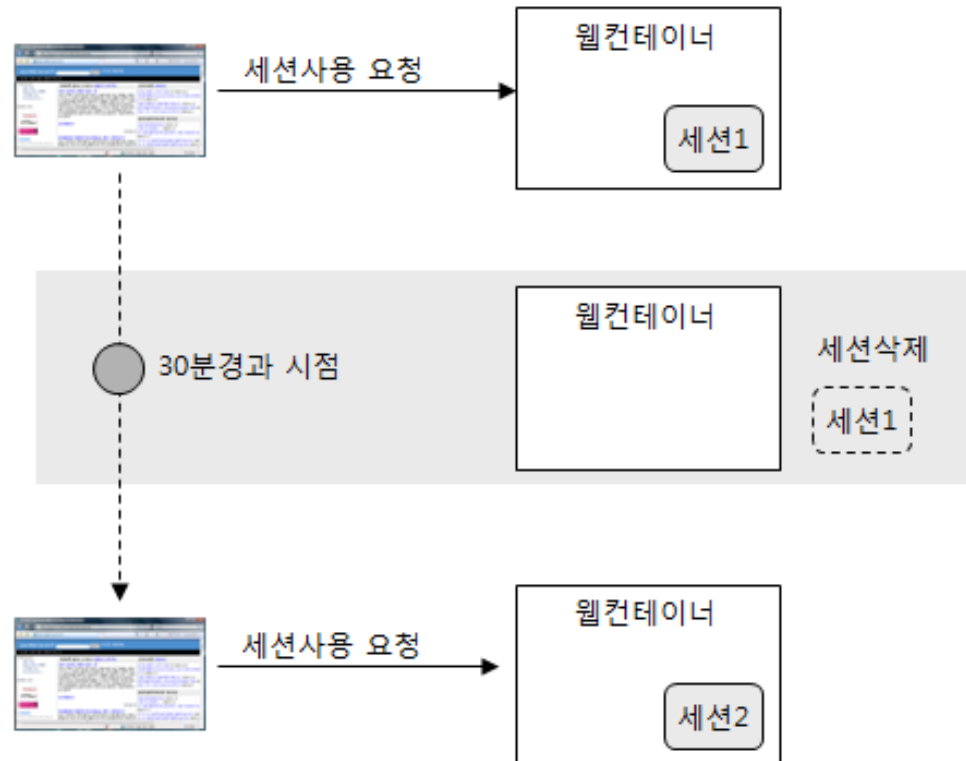
## ■ 세션 종료 / 타임 아웃

- `session.invalidate()` 을 이용해서 세션 종료
  - 세션이 종료되면 기존에 생성된 세션이 삭제
  - 이후 접근 시 새로운 세션 생성 됨
- 마지막 세션 사용 이후 유효 시간이 지나면 자동 종료

- `web.xml` 파일에서 지정

```
<session-config>  
  <session-timeout>  
    30  
  </session-timeout>  
</session-config>
```

- 또는 `session` 기본 객체의 `setMaxInactiveInterval()` 메서드 이용해서 지정
  - 초 단위 지정



## ■ 세션 종료 / 타임 아웃

### WcloseSession.jsp

```
<%@ page contentType = "text/html; charset=euc-kr" %>  
<%
```

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

```
%>
```

```
<html>
```

```
<head> <title> 세션 종료 </title> </head>
```

```
<body>
```

세션을 종료하였습니다.

```
</body>
```

```
</html>
```

# 세션을 이용한 인증 처리



## ■ 세션 종료 / 타임 아웃

### WmemberWsessionLoginForm.jsp

```
<%@ page contentType = "text/html; charset=euc-kr" %>
<html>
<head> <title>로그인폼</title> </head>
<body>

<form action="<%= request.getContextPath() %>/member/sessionLogin.jsp"
        method="post">
아이디 <input type="text" name="id" size="10">
암호 <input type="password" name="password" size="10">
<input type="submit" value="로그인">
</form>

</body>
</html>
```

### ~~W~~member~~W~~sessionLogin.jsp

```
<%@ page contentType = "text/html; charset=euc-kr" %>
<%
    String id = request.getParameter("id");
    String password = request.getParameter("password");

    if (id.equals(password)) {
        session.setAttribute("MEMBERID", id);
    }
%>
<html>
<head> <title> 로그인성공 </title> </head>
<body>

로그인에 성공했습니다.

</body>
</html>
```

### WmemberWsessionLogin.jsp

```
<%  
        } else { // 로그인 실패시  
%>  
<script>  
alert("로그인에 실패하였습니다.");  
history.go(-1);  
</script>  
<%  
        }  
%>
```

## ■ 세션 종료 / 타임 아웃

### ~~W~~memberWsessionLoginCheck.jsp

```
<%@ page contentType = "text/html; charset=euc-kr" %>
```

```
<%
```

```
    String memberId = (String)session.getAttribute("MEMBERID");
```

```
    boolean login = memberId == null ? false : true;
```

```
%>
```

```
<html>
```

```
<head> <title>로그인여부 검사</title> </head>
```

```
<body>
```

```
<%      if (login) {          %>
```

아이디 "<%= memberId %>"로 로그인 한 상태

```
<%      } else { %>
```

로그인하지 않은 상태

```
<%      }          %>
```

```
</body>
```

```
</html>
```

## ■ 세션 종료 / 타임 아웃

### WmemberWsessionLogout.jsp

```
<%@ page contentType = "text/html; charset=euc-kr" %>
<%
```

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

```
%>
```

```
<html>
```

```
<head> <title> 로그아웃 </title> </head>
```

```
<body>
```

로그아웃하였습니다.

```
</body>
```

```
</html>
```

## ■ session 기본 객체에 저장될 값 타입

- 필요한 값을 개별 속성으로 저장하는 것 보다는
  - 유지 보수 불편함 (신규 속성 추가 시 변경할 코드 증가 등)
- 속성을 하나로 묶은 자바 클래스를 작성해서 저장하는 것이 좋음
  - 유지 보수 편리함
- 클래스를 구성해 보자.

## ■ 로그인 페이지 만들기

- 로그인 페이지를 완성하시오.