

JSP 프로그래밍

(재)대덕인재개발원

10 클라이언트와의 대화 2 - 세션

- ▶ 1. 세션 사용하기 : session 기본 객체
- ▶ 2. 세션을 사용한 인증 정보 유지
- ▶ 3. 연관된 정보 저장을 위한 클래스 작성

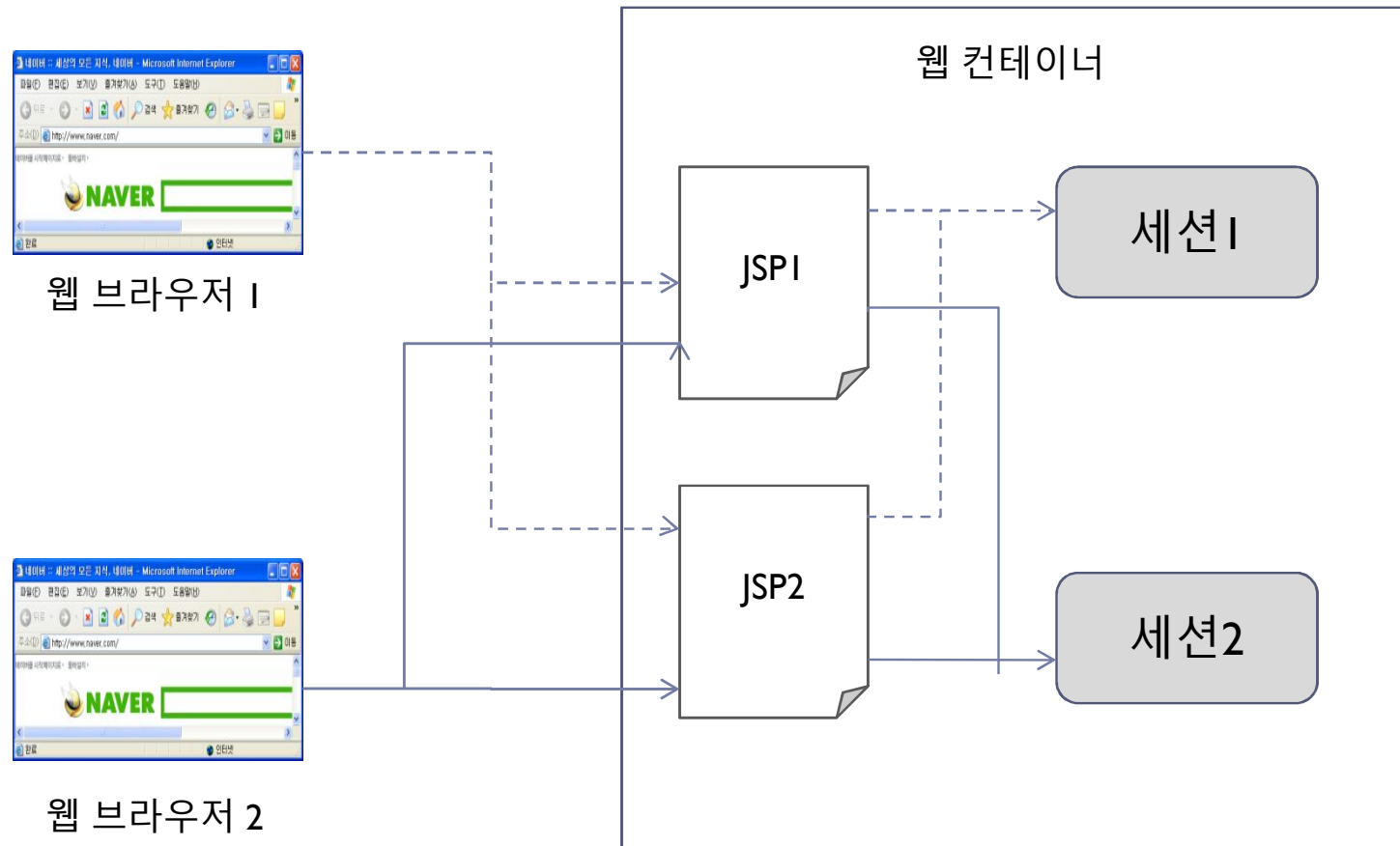


10.1 세션 사용하기 : session 기본 객체

- ▶ 세션은 웹 브라우저가 아니라 서버에 값을 저장된다는 점이 쿠키와 차이점이다.
- ▶ 쿠키가 웹 브라우저에서 정보를 보관할 때 사용된다면, 세션은 웹 컨테이너에서 정보를 보관할 때 사용된다.
- ▶ 세션은 오직 서버에서만 생성된다.
- ▶ 웹 컨테이너는 기본적으로 하나의 웹 브라우저에 하나의 세션을 생성한다.
- ▶ 세션이 생성되면 session 기본 객체를 통해서 세션을 사용할 수 있다.



10.1 세션 사용하기 : session 기본 객체



[그림] 세션은 하나의 웹 브라우저와 연관된 서버 영역의 저장 공간이다.

10.1.1 세션 생성하기

- ▶ JSP에서 세션을 생성하기 위해서는 page 디렉티브의 session 속성을 “true”로 지정해 주면 된다.
- ▶ 세션은 웹 브라우저가 최초로 접속할 때에 생성되며 그 이후로는 이미 생성된 세션을 사용한다.

```
<%@ page contentType="text/html; charset=utf-8" %>  
<%@ page session="true" %>  
<%  
    session.setAttribute("userInfo", userInfo);  
%>
```



10.1.2 session 기본 객체

- ▶ 웹 브라우저마다 별도의 세션을 갖게 되는데, 이를 구분하기 위해서 세션ID를 할당한다.
- ▶ 웹 컨테이너는 웹 브라우저에 세션 ID를 전송하며, 웹 브라우저는 웹 컨테이너에 연결할 때마다 매번 세션ID를 알려준다.
- ▶ session 객체가 제공하는 세션 정보 관련 메서드

메서드	리턴 타입	설 명
getId()	String	세션의 고유 ID를 구한다.(세션 ID라고 한다.)
getCreationTime()	long	세션이 생성된 시간을 구한다. 시간은 1970년 1월 1일 이후 흘러간 시간을 의미하며, 단위는 1/1000초이다.
getLastAccessedTime()	long	웹 브라우저가 가장 마지막에 세션에 접근한 시간을 구한다. 최근 접근 시간을 기록함으로써 세션의 타임아웃을 관리하기 위함이다. 시간은 1970년 1월 1일 이후 흘러간 시간을 의미하며, 단위는 1/1000초이다.

10.1.3 session 객체의 속성 사용

- ▶ 세션은 지정된 유효 시간 동안 유지되므로, 웹 어플리케이션을 실행하는 동안 지속적으로 사용해야 하는 데이터의 저장 장소로서 세션이 적당하다.
- ▶ 세션에 값을 저장할 때는 속성을 사용한다.

```
<%@ page contentType="text/html; charset=utf-8" %>
<%
    session.setAttribute("MEMBERID", "san");
    session.setAttribute("NAME", "이산");
%>
<html>
<head><title> 세션에 정보 저장 </title></head>
<body>
세션에 정보를 저장하였습니다.<br>
<%
    String name = (String)session.getAttribute("NAME");
%>
회원명 : <%= name %>
</body>
</html>
```

10.1.4 세션 종료

- ▶ session.invalidate() 메소드를 사용하여 세션을 종료하면 된다.

```
<%@ page contentType="text/html; charset=utf-8" %>
<%
    session.invalidate();
%>

<html>
<head><title> 세션 종료 </title></head>

<body>
세션을 종료하였습니다.
</body>
</html>
```


10.1.5 세션의 유효 시간

- ▶ 세션은 마지막 접근 시간으로부터 일정 시간 이내에 다시 세션에 접근하지 않을 경우 자동으로 세션을 종료하는 기능을 갖고 있다.
- ▶ `<session-timeout>`의 값을 0이나 음수로 설정하면 세션은 유효 시간을 갖지 않는다.

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee" version="2.5">  
  <session-config>  
    <session-timeout>30</session-timeout>  
  </session-config>  
</web-app>
```

```
<%  
  // 초 단위로 설정  
  session.setMaxInactiveInterval(60 * 60);  
%>
```

10.1.6 request.getSession()을 이용한 세션 생성

- ▶ request 기본 객체의 getSession() 메서드를 이용해서 HttpSession을 생성할 수 있다.
- ▶ request.getSession() 메서드는 현재 요청과 관련된 session 객체를 리턴해 준다.

```
<%@ page session="false" %>
<%
// session이 존재하지 않으면
// 새로운 session 객체를 생성하여 리턴한다.
HttpSession httpSession = request.getSession();

List list = (List)httpSession.getAttribute("list");
list.add(productId);
%>
```

```
<%@ page session="false" %>
<%
// session이 존재하지 않으면 null 리턴
HttpSession httpSession = request.getSession(false);

List list = null;
if(httpSession != null){
    list = (List)httpSession.getAttribute("list");
}else{
    list = Collection.emptyList();
}
%>
```

10.2 세션을 사용한 인증 정보 유지

- ▶ 세션을 사용한 로그인 기법은 쿠키와 비슷하며, 일반적인 방식은 다음과 같다.
 - ▶ 1. 로그인에 성공하면 `session` 객체의 특정 속성에 데이터를 기록한다.
 - ▶ 2. 이후로 `session` 객체의 특정 속성이 존재하면 로그인한 것으로 간주한다.
 - ▶ 3. 로그아웃 할 경우 `session.invalidate()` 메서드를 호출하여 세션을 종료한다.



10.2.1 인증된 사용자 정보 session 객체에 저장하기

```
<%@ page contentType="text/html; charset=utf-8" %>
<%
    String id = request.getParameter("id");
    String password = request.getParameter("password");
    if(id.equals(password)){
        session.setAttribute("MEMBERID", id);
    }
%>
<html>
<head><title> 로그인 성공 </title></head>
<body>
로그인에 성공하였습니다.
</body>
</html>
<%
    }else{ // 로그인 실패시
%>
<script>
    alert("로그인에 실패하였습니다.");
    history.go(-1);
</script>
<%
    }
%>
```

10.2.2 인증 여부 판단

```
<%@ page contentType="text/html; charset=utf-8" %>
<%
    String memberId = (String)session.getAttribute("MEMBERID");
    boolean login = memberId == null ? false : true;
%>
<html>
<head><title> 로그인 여부 검사 </title></head>
<body>
<%
    if(login){
%>
        아이디 "<%= memberId %>"로 로그인 한 상태
<%
    }
    else{
%>
        로그인하지 않은 상태
<%
    }
%>
</body>
</html>
```

10.2.3 로그아웃 처리

```
<%@ page contentType="text/html; charset=utf-8" %>
<%
    // 세션 제거
    session.invalidate();

    // 혹은 로그인 관련 속성 제거
    session.removeAttribute("MEMBERID");
%>
<html>
<head><title> 로그아웃 </title></head>
<body>

로그아웃하였습니다.

</body>
</html>
```

