Contents

❖ 쿠키와 세션

- HTTP와 연결 상태 유지
- 쿠기와 세션
- 쿠키(Cookie)
- 세션(Session)

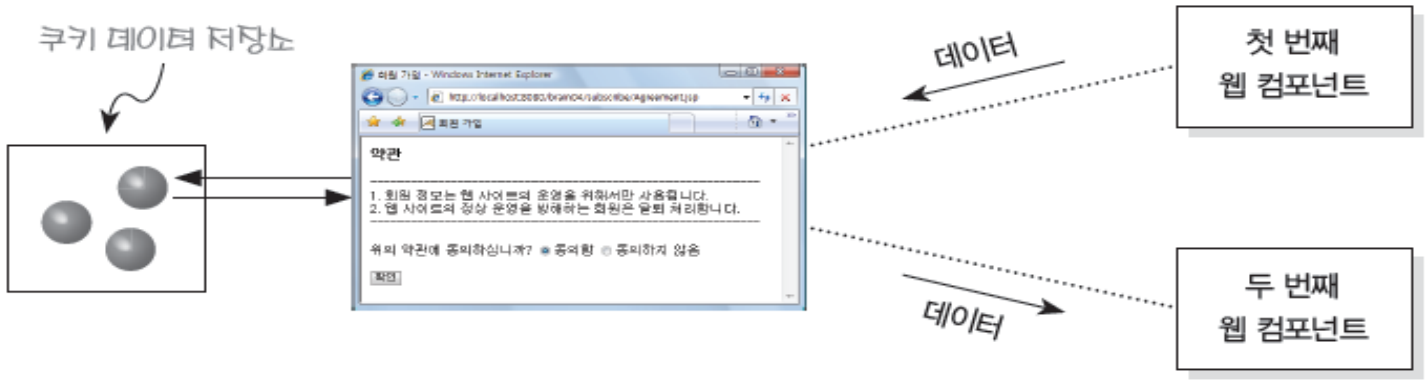
1. HTTP와 연결 상태 유지

❖ HTTP의 연결 동작(Stateless)

- HTTP 통신은 TCP/IP 기반의 비연결지향 프로토콜로 연결 지속성이 없다.
- HTTP에서 연결 상태의 지속성을 유지하기 위해 필요한 방법이 쿠키(Cookie)와 세션(Session) 기술이다.

2. 쿠키와 세션

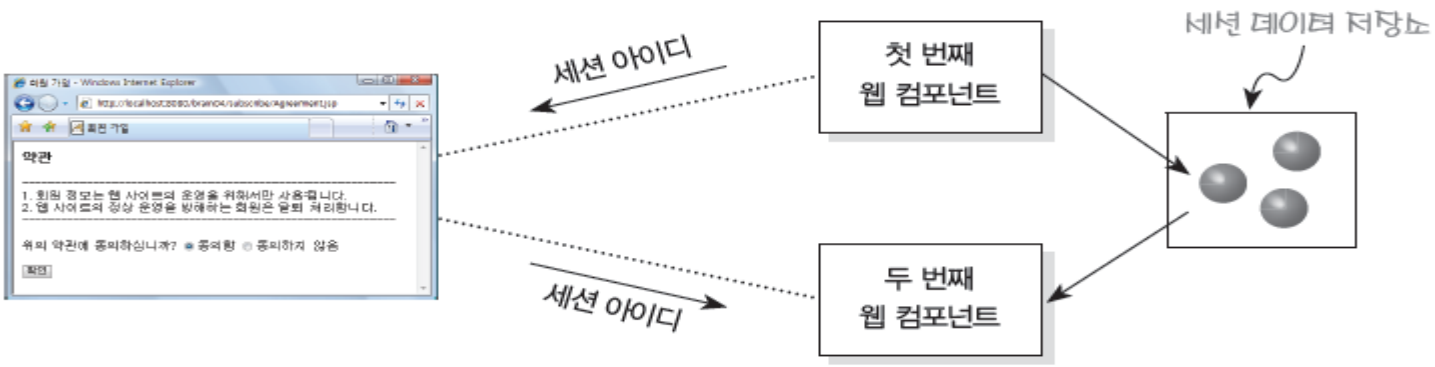
- 쿠키 기술은 웹 서버가 웹 브라우저로 데이터를 보냈다가 웹 서버 쪽으로 다시 되돌려 받는 방법을 사용한다.
- 첫 번째 웹 컴포넌트는 웹 브라우저로 HTML 문서를 보낼 때 쿠키를 함께 보내고 웹 브라우저는 그 쿠키를 저장해 두었다가 두 번째 웹 컴포넌트를 호출할 때 URL과 함께 웹 서버로 보낸다.



쿠키 기술을 이용한 웹 컴포넌트 간의 데이터 전달

2. 쿠키와 세션

- 세션 기술은 웹 브라우저를 거치지 않고 웹 서버에 있는 특정 데이터 영역을 통해 데이터를 전달하는 방법이다.
- 첫 번째 웹 컴포넌트는 세션을 생성하여 웹 서버 쪽에 저장해 놓고 세션 아이디만 웹 브라우저로 보낸다. 웹 브라우저는 세션 아이디를 쿠키로 저장해 두었다가 두 번째 웹 컴포넌트를 호출할 때 요청 데이터와 함께 웹 서버로 보내고 웹 서버는 세션 아이디를 이용해 세션 저장소에 저장된 세션 데이터를 찾을 수 있다.

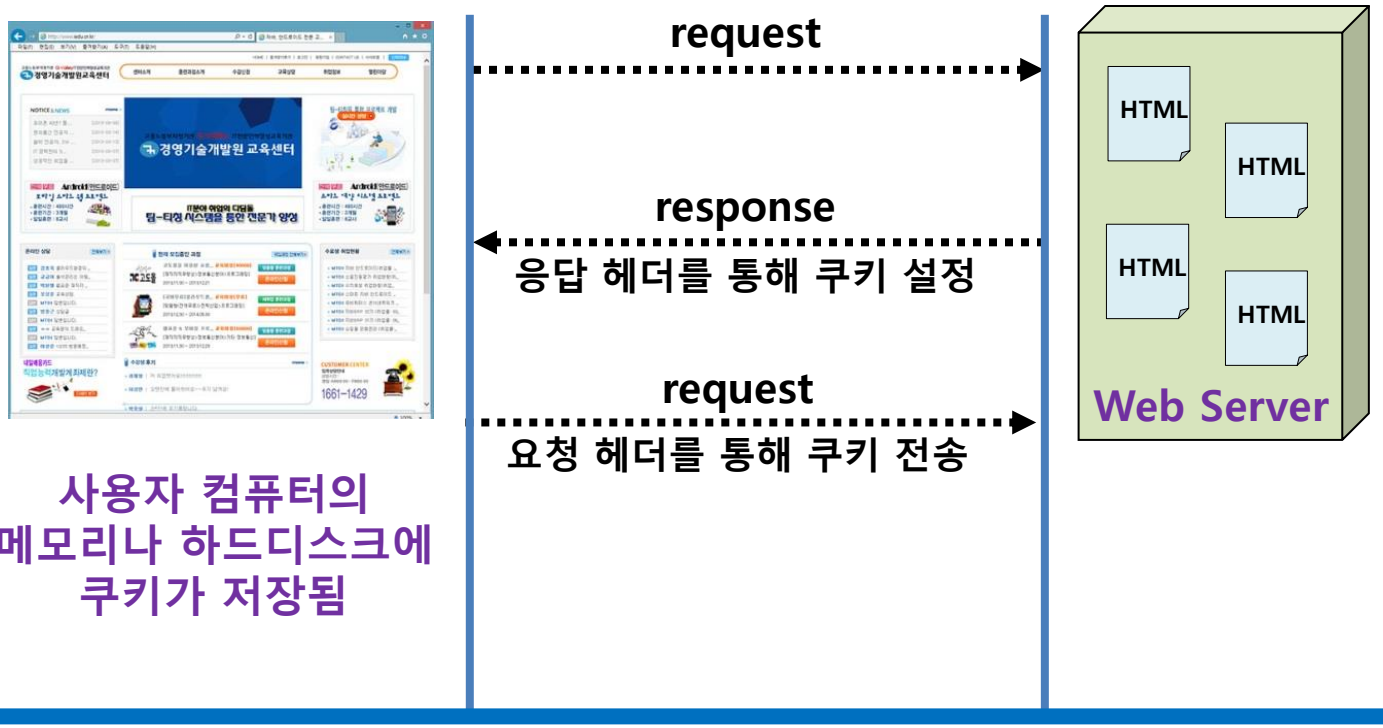


세션 기술을 이용한 웹 컴포넌트 간의 데이터 전달

2. 쿠키(Cookie)

❖ 쿠키(Cookie)의 동작방식

- 웹 브라우저의 요청에 대한 응답 데이터에 쿠키를 설정하면 웹 브라우저는 그 쿠키를 클라이언트의 쿠키저장소(메모리나 하드디스크)에 저장한다.
- 웹 브라우저는 저장된 쿠키 정보를 다음 요청의 헤더에 포함하여 서버로 전송한다.
- 웹 서버는 요청 헤더의 쿠키정보를 이용하여 클라이언트를 식별할 수 있다.



2. 쿠키(Cookie)

❖ 구성 요소

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Set-Cookie: name=%ED%99%8D%EA%B8%B8%EB%8F%99; Expires=Mon, 28-Jan-2019 18:21:19 GMT
Content-Type: text/html;charset=UTF-8
Content-Length: 103
Date: Mon, 28 Jan 2019 18:18:19 GMT
```

```
GET /JSPStudyCh08/jspCookieHangulCreation.jsp HTTP/1.1
Host: localhost:8080
... 중략...
Connection: Keep-Alive
Cookie: name=%ED%99%8D%EA%B8%B8%EB%8F%99
```

← 쿠키 설정 후 요청 헤더

❖ 쿠키 이름의 제약

- **쿠키의 이름은 아스키 코드의 알파벳과 숫자만을 포함할 수 있다.**
- **콤마(,), 세미콜론(;), 공백(' ') 등의 문자는 포함할 수 없다.**
- **'$'로 시작할 수 없다.**
- **URLEndoder.encode()와 URLDecoder.decode()를 이용하면 한글 가능**

2. 쿠키(Cookie)

❖ 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.

2. 쿠키(Cookie)

- 기본적으로 쿠키는 한글과 같은 문자를 사용할 수 없음
  - 쿠키의 값을 URLEncoding 기법을 통해 지정하면 가능

- 쿠키 값 한글 처리
  - 값 설정시 : URLEncoder.encode("값", “utf-8")

```
new Cookie("name", URLEncoder.encode("값", "euc-kr"));
```

  - 값 조회시 : URLDecoder.decode("값", "euc-kr")

```
Cookie[] cookies = request.getCookies();
for(int i = 0; cookies.length; i++) {
    String value = URLDecoder.decode(cookie[i].getValue(), "euc-kr");
}
```

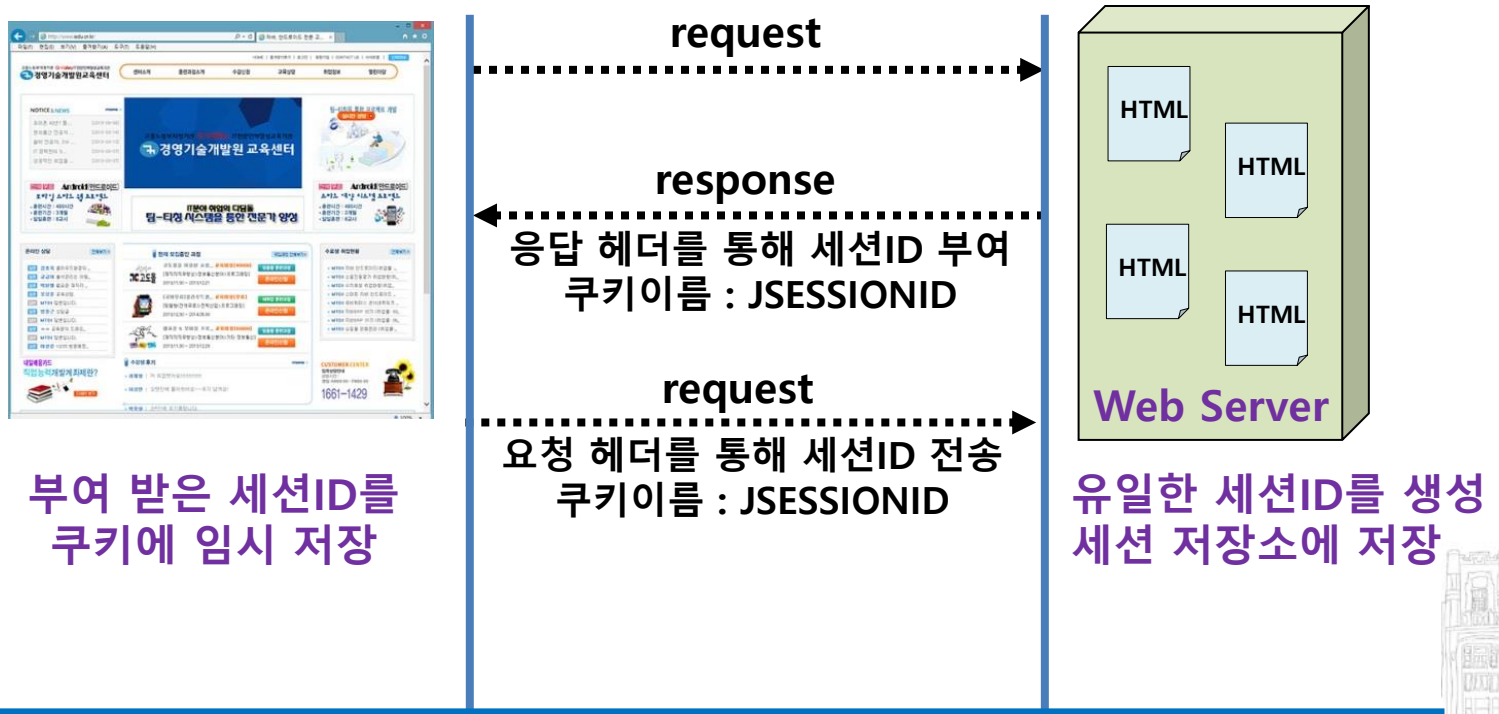
2. 쿠키(Cookie)

- 도메인을 지정하면 해당 도메인에 쿠키 전달
  - Cookie.setDomain(“.jspstudy.com”)으로 쿠키 설정
    - .jspstudy.com – . 으로 시작하는 경우 관련 도메인에 모두 쿠키 전송
      – jsp.jspstudy.com, study.jspstudy.com 등등
    - www.jspstudy.com – 특정 도메인에 대해서만 쿠키를 전송
  - 웹 브라우저가 접속한 도메인과 다르면 쿠키를 저장하지 않음
- 경로 설정시 해당 경로를 기준으로 쿠키 전달
  - 경로 미 설정시는 요청 URL의 경로에 대해서만 쿠키 전달
  - 경로 설정시 설정한 경로 및 그 하위 경로에 대해서 쿠키 전달
  - Cookie.setPath(“/JSPStudyCh09”)로 경로 설정 – 폴더 단위 설정
- 유효 기간
  - 유효 기간을 지정하지 않으면 웹 브라우저가 닫을 때 쿠키도 함께 삭제
  - Cookie.setMaxAge(60 * 5)로 초단위 쿠키 유효 시간 설정
    - 유효 기간이 지나지 않을 경우 웹 브라우저를 닫더라도 쿠키가 삭제되지 않고
      이후 웹 브라우저가 요청할 때 마다 요청 헤더에 포함하여 해당 쿠키를 전송

3. 세션(Session)

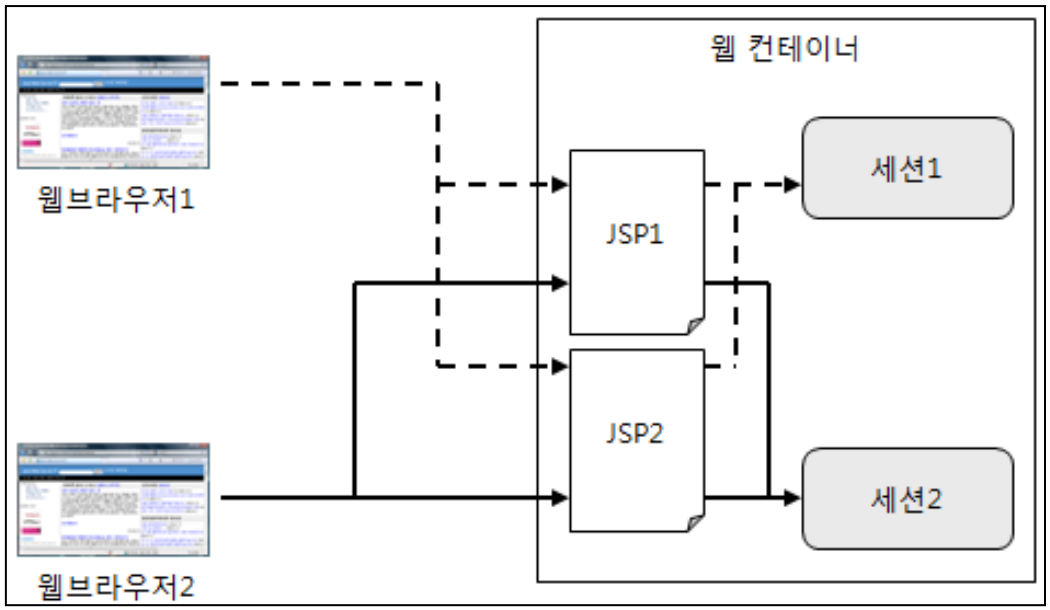
❖ 세션(Session)의 동작방식

- 웹 브라우저의 첫 요청에 대한 응답 데이터에 세션ID를 생성하고 웹 브라우저에게 세션ID를 부여한다. 이때 생성한 세션 ID는 서버의 세션 저장소에 저장된다.
- 웹 브라우저는 부여 받은 세션ID를 다음 요청의 헤더에 포함하여 서버로 전송한다.
- 웹 서버는 요청 헤더의 세션ID를 이용하여 클라이언트를 식별할 수 있다.



3. 세션(Session)

- 웹 컨테이너에서 클라이언트의 정보를 보관할 때 사용
- 세션은 오직 서버에서만 생성하고 관리
- 브라우저 종류마다 또는 클라이언트마다 세션을 생성



3. 세션(Session)

❖ 서블릿 클래스에서 세션 사용하기

- 서블릿 클래스에서 세션을 시작하기 위해 doGet(), doPost()의 첫 번째 파라미터로 받은 HttpServletRequest 객체의 getSession()으로 세션 객체를 얻는다.
- getSession()은 진행 중인 세션이 없을 때는 새로운 세션을 시작하고, 진행 중인 세션이 있는 경우 그 세션 정보를 HttpSession 객체로 만들어서 리턴한다.
- 세션은 같은 웹 애플리케이션 컴포넌트 끼리 데이터 교환이 가능하다.
- setAttribute()로 세션 데이터 영역에 데이터를 저장한다.
- setAttribute()는 동일한 영역에 같은 이름의 데이터가 있으면 덮어 쓰기 한다.

```
HttpSession session = request.getSession();
```

```
session.setAttribute();
```

- 세션사용이 끝나면 removeAttribute()를 이용해 데이터를 삭제 하거나 invalidate()를 이용해 세션을 종료하고 새로운 세션을 시작할 수 있다.

```
session.removeAttribute();
```

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

3. 세션(Session)

❖ JSP 페이지에서 세션 사용하기

- page 디렉티브의 session 속성을 false로 지정하지 않는 한 session 내장변수로 접근
- session 내장변수를 사용해 setAttribute()로 세션 영역에 데이터를 저장하고 getAttribute()로 세션 영역의 데이터를 읽어 온다.

```
session.setAttribute();
```

```
String str = (String) session.getAttribute();
```

- 세션사용이 끝나면 session 내장 변수의 removeAttribute()를 이용해 데이터를 삭제 하거나 invalidate()를 이용해 세션을 종료

```
session.removeAttribute();
```

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

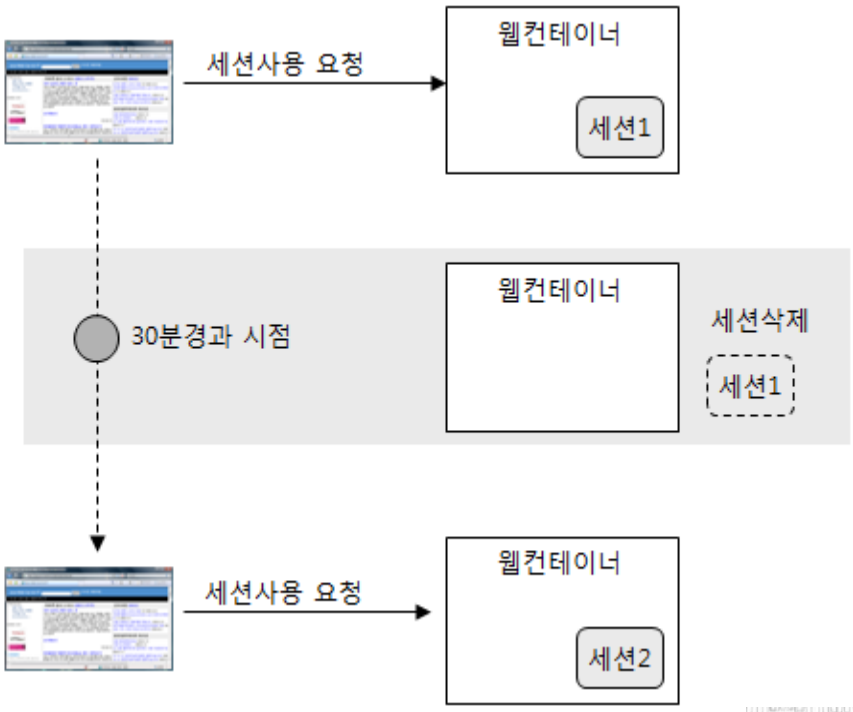
3. 세션(Session)

- **session.invalidate() 을 이용해서 세션 종료**
  - 세션이 종료되면 기존에 생성된 세션이 삭제
  - 이후 접근 시 새로운 세션이 생성 됨

- **마지막 세션 사용 후 지정한 시간이 지나면 자동 종료**
  - web.xml 파일에서 지정(분 단위)

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

  - session 내장 변수의 setMaxInactiveInterval()를 이용해 초 단위로 지정 가능



3. 세션(Session)

❖ 쿠키와 세션의 차이점

구 분	쿠 키	세 션
저 장	클라이언트의 메모리나 디스크	서버의 세션 저장소
데이터 형식	텍스트	Object
유효시간	쿠키 저장 시 설정 가능 따로 설정하지 않으면 브라우저를 종료할 때 까지	브라우저를 종료할 때까지, 설정한 시간 내에 새로운 요청이 없을 때까지 유효
용 량	한 도메인당 20개 쿠키 1개당 4kb 총 300개	서버의 리소스 범위 내