

```

1 Servlet Cookie
2 1. Cookie 클래스를 이용한 쿠키 처리 방법
3     1) 쿠키는 사용자의 상태 정보를 클라이언트에서 관리하는 메커니즘을 의미
4     2) 클라이언트에 정보가 저장되기 때문에 서버의 부하가 크지 않지만 보안에 매우 취약
5     3) 쿠키는 웹 사이트의 도메인당 300개까지 저장 가능
6     4) 클라이언트에서 쿠키를 사용하지 못하도록 설정할 수 있기 때문에 제약이 있음
7     5) 쿠키는 클라이언트의 브라우저 메모리 및 OS 파일에 저장 가능
8     6) 기본 저장은 브라우저 메모리이기 때문에 브라우저를 종료하면 자동으로 쿠키 정보도 제거
9     7) 쿠키 만료시간은 setMaxAge(sec) 메서드를 사용하며 지정된 시간까지 OS 파일에 저장
10
11
12 2. Constructor
13     1) javax.servlet.http.Cookie;
14     2) public Cookie(java.lang.String name, java.lang.String value)
15         Cookie(name, value) : 쿠키 생성시 사용되는 생성자
16
17
18 3. Methods
19     1) response.addCookie(cookie) : 생성된 쿠키를 응답처리
20     2) request.getCookies() : 클라이언트로부터 쿠키 정보를 배열로 리턴
21     3) setMaxAge(second) : 쿠키 만료시간을 설정
22     4) getMaxAge() : 쿠키 만료시간을 얻음
23     5) getName() : 쿠키 이름을 얻음
24     6) setValue(value) : 쿠키 값을 설정
25     7) getValue() : 쿠키 값을 얻음
26     8) length: 쿠키 개수를 리턴
27
28
29 4. Lab1
30     1) CookieGet.java
31
32     public class CookieGet extends HttpServlet {
33         public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
34             IOException {
35             response.setContentType("text/html;charset=utf-8");
36             PrintWriter out = response.getWriter();
37             out.println("<html><body>");
38             out.println("<h1>Cookie 추출</h1><hr>");
39             Cookie [] cookies = request.getCookies();
40             if(cookies != null && cookies.length > 0){
41                 out.println("다음과 같은 Cookie 정보가 전달 되었습니다.<br>");
42                 for(int i = 0 ; i < cookies.length; i++){
43                     out.print("Cookie Name : " + cookies[i].getName() + "<br>");
44                     out.println("Cookie Value : " + cookies[i].getValue() + "<br>");
45                 }
46             }else
47                 out.println("전달된 cookie 정보가 없습니다.");
48             out.println("</body></html>");
49             out.close();
50         }
51     }
52
53     2) CookieSet.java
54
55     public class CookieSet extends HttpServlet {
56         protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
57             ServletException, IOException {
58             Cookie c1 = new Cookie("discount", "20");
59             Cookie c2 = new Cookie("item", "sports");
60             response.setContentType("text/html;charset=utf-8");
61             response.addCookie(c1);
62             c2.setMaxAge(60 * 60 * 24 * 3);
63             response.addCookie(c2);
64             PrintWriter out = response.getWriter();
65             out.println("<html><body>");
66             out.println("<h1>Cookie 설정</h1><hr>");
67             out.println("설정 완료!!!");
68             out.println("</body></html>");
69             out.close();
70         }
71     }
72
73 5. Lab2

```

### 1)CartBasketCookieServlet.java

```
public class CartBasketCookieServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
        IOException {
        doPost(request, response);
    }
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
        IOException {
        response.setContentType("text/html; charset=utf-8");
        PrintWriter out = response.getWriter();
        out.print("<html><body>");
        out.print("장바구니 리스트<br>");
        //기존 쿠키 얻기
        Cookie [] cookies = request.getCookies();
        if( cookies != null ){
            for (Cookie c : cookies) {
                out.print(c.getName() + " : " + c.getValue() + "<br>");
            }
        }else{
            out.print("장바구니 비었음<br>");
        }
        out.print("<a href='product.html'>상품 선택 페이지</a><br>");
        out.print("<a href='CartDeleteCookie'>장바구니 비우기</a><br>");
        out.print("</body></html>");
    }
}
```

### 2)product.html

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>상품 리스트</title>
</head>
<body>
<h1>상품 리스트</h1>
<form action="/servlet/CartSaveCookie" method="get">
<input type="radio" name="product" value="BMW">BMW<br>
<input type="radio" name="product" value="SM5">SM5<br>
<input type="radio" name="product" value="K7">K7<br>
<input type="submit" value="상품저장">
</form>
</body>
</html>
```

### 3)CartSaveCookieServlet.java

```
public class CartSaveCookieServlet extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
        IOException {
        doPost(request, response);
    }
    public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
        IOException {
        response.setContentType("text/html; charset=utf-8");
        PrintWriter out = response.getWriter();
        // 입력 파라미터 얻기
        String product = request.getParameter("product");
        //기존 쿠키 얻기
        Cookie [] cookies = request.getCookies();
        Cookie c = null;
        if(cookies == null || cookies.length == 0){
            c = new Cookie("product" , product );
        }else{
            c = new Cookie("product"+ ( cookies.length+1 ) , product );
        }
        //쿠키를 응답처리
        // c.setMaxAge(60*60);
        response.addCookie(c);
        out.print("<html><body>");
        out.print("Product 추가");
        //out.print("<a href='/servlet/CartBasketCookie'>장바구니 보기</a>");
    }
}
```

```

143         out.print("<a href="+response.encodeURL("CartBasketCookie")+ ">장바구니 보기2</a>");
144         out.print("</body></html>");
145     }
146 }
147

```

#### 4)CartDeleteCookieServlet.java

```

150 public class CartDeleteCookieServlet extends HttpServlet {
151     public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
        IOException {
152         doPost(request, response);
153     }
154     public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
        IOException {
155         response.setContentType("text/html; charset=utf-8");
156         PrintWriter out = response.getWriter();
157         out.print("<html><body>");
158         out.print("장바구니 비웠음!!");
159         //기존 쿠키 얻기
160         Cookie [] cookies = request.getCookies();
161         if( cookies != null ){
162             for (Cookie c : cookies) {
163                 c.setMaxAge(1);
164                 response.addCookie(c);
165             }
166         }else{
167             out.print("장바구니 비웠음<br>");
168         }
169         out.print("<a href='product.html'>상품 선택 페이지</a><br>");
170         out.print("</body></html>");
171     }
172 }
173

```

---

## Session Tracking

1. http 프로토콜은 stateless protocol 즉 상태를 저장하지 않는 프로토콜이다.
2. 클라이언트가 웹 서버에 요청하고 응답받으면 즉시 연결을 끊는 방식
3. HTTP 프로토콜의 문제점을 극복하기 위해서 사용자의 상태 정보를 관리하는 메커니즘이 필요 → '세션 관리'
4. Session Tracking(세션 관리)의 방법
  - HttpSession 클래스를 이용한 세션 처리 방법
  - Cookie 클래스를 이용한 쿠키 처리 방법

## Session

1. HttpSession 클래스를 이용한 세션 처리 방법
  - 1) 세션(Session)이란 사용자의 상태 정보를 서버에서 관리하는 메커니즘을 의미
  - 2) 세션의 정보는 클라이언트가 서버에 접속해서 종료될 때까지(브라우저를 종료할 때까지) 유지
  - 3) 상태 정보가 서버에 저장되기 때문에 서버의 부하가 클 수 있음
  - 4) time-out 제한을 두어 일정시간(기본 30분) 동안 요청이 없으면 서버는 세션 정보를 유지하지 않고 제거
2. Session 사용시 많이 사용되는 메소드 목록
  - 1) request.getSession() :
    - session 영역을 얻을 때 사용하며 session 영역이 없으면 새로 생성하고, 있으면 생성된 영역을 참조
    - request.getSession(true) 메서드와 동일한 기능
    - 일반적으로 세션을 처음 생성하는 서블릿에서 지정
  - 2) request.getSession(false)
    - session 영역을 얻을 때 사용하며 session 영역이 없으면 null 값을 리턴하고, 있으면 생성된 영역을 참조
    - 일반적으로 세션을 사용하는 서블릿에서 주로 사용
  - 3) sess.getId() : 생성된 세션 ID 값을 반환
  - 4) sess.setAttribute(key,value)
    - session 영역에 속성(Attribute) 값을 설정
    - Session scope를 따르기 때문에 브라우저를 종료하기 전까지 사용 가능
    - 단, 기본으로 30분 이상 요청이 없을 시 제거
  - 5) sess.getAttribute(key) : session 영역에 저장된 속성 값을 반환
  - 6) sess.removeAttribute(key) : session 영역에 저장된 속성을 제거
  - 7) setMaxInactiveInterval(sec) : 지정된 시간만큼 세션을 유지. 단위는 초(second)
  - 8) getMaxInactiveInterval() : 세션의 유지시간을 반환
  - 9) sess.invalidate() : 즉시 session 영역을 제거. 로그아웃 기능 시 사용
3. 세션을 제거하는 방법
  - 1) time-out 을 지정하여 제거하는 방법

```

215 - setMaxInactiveInterval(second) 메서드 사용
216 session.setMaxInactiveInterval( 60*60*24 ); //24시간 유지
217 - web.xml 사용
218 <web-app>
219     <!-- 단위는 분(minute) -->
220     <session-config>
221         <session-timeout>60</session-timeout>
222     </session-config>
223 </web-app>
224 2) 즉시 제거하는 방법
225 - invalidate( ) 메서드 사용
226 session.invalidate( ); // 즉시 제거
227 3) 세션에 저장된 특정 속성 값을 제거하는 방법
228 - removeAttribute(name) 메서드 사용
229 session.removeAttribute(name); //name 값에 해당되는 속성 값만 제거

```

#### 4. Lab1

##### 1)SessionTestServlet.java

```

235 public class SessionTestServlet extends HttpServlet {
236     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
237         response.setContentType("text/html;charset=utf-8");
238         PrintWriter out = response.getWriter();
239         HttpSession session = null;
240         String msg = null;
241         String param = request.getParameter("comm");
242         if(param.equals("create")){
243             session = request.getSession();
244             if(session.isNew()){
245                 msg = "세션 객체가 새로 생성됐습니다";
246             }else
247                 msg = "이미 세션 객체가 생성되어 있습니다";
248         }else if(param.equals("destroy")){
249             session = request.getSession(false);
250             if(session != null){
251                 session.invalidate();
252                 msg = "세션 객체가 삭제됐습니다.";
253             }else
254                 msg = "전달된 명령을 해석할 수 없습니다";
255         }
256         out.println("<html><body><h3>처리결과 : </h3>" + msg);
257         out.println("</body></html>");
258         out.close();
259     }
260 }

```

##### 2)CountTestServlet.java

```

264 public class CountTestServlet extends HttpServlet {
265     protected void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
266         response.setContentType("text/html;charset=utf-8");
267         PrintWriter out = response.getWriter();
268         out.println("<html><body bgcolor='yellow'");
269         out.println("<h2>누적시킬 숫자를 입력하세요...</h2>");
270         out.println("<form method='post' action='\"/DemoWeb/CountTestServlet\">");
271         out.println("<p><input type='\"text\"' name='\"number\"'></p>");
272         out.println("<input type='\"submit\"' name='\"see\"' value='\"전송\"'>");
273         out.println("<input type='\"submit\"' name='\"buy\"' value='\"재입력\"'>");
274         out.println("</form></body></html>");
275         out.close();
276     }
277     protected void doPost(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
278         HttpSession session = request.getSession(true);
279         if(session.isNew())
280             session.setAttribute("number", new int[]{0});
281         int [] sum = (int [])session.getAttribute("number");
282         int number = Integer.parseInt(request.getParameter("number"));
283         sum[0] += number;
284         response.setContentType("text/html;charset=euc-kr");
285         PrintWriter out = response.getWriter();

```

```

286         out.println("<html><body><h3>누적된 값 : " + sum[0] + "</h3>");
287         out.println("<h3>세션 ID : " + session.getId() + "</h3>");
288         out.println("<h3>세션발생시간 : " + new java.util.Date(session.getCreationTime()).toString() + "</h3>");
289         out.println("<hr><a href=" + request.getRequestURI() + ">이전페이지</a>");
290         out.println("</body></html>");
291         out.close();
292     }
293 }

```

## 5. Lab2

### 1)product.html

```

299 <!DOCTYPE html>
300 <html>
301     <head>
302         <meta charset="utf-8">
303         <title>상품 리스트</title>
304     </head>
305     <body>
306         <h1>상품 리스트</h1>
307         <form action="/servlet/CartSaveCookie" method="get">
308             <input type="radio" name="product" value="BMW">BMW<br>
309             <input type="radio" name="product" value="SM5">SM5<br>
310             <input type="radio" name="product" value="K7">K7<br>
311             <input type="submit" value="상품저장">
312         </form>
313     </body>
314 </html>

```

### 2)CartSaveServlet.java

```

317
318 public class CartSaveServlet extends HttpServlet {
319     public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
        IOException {
320         doPost(request, response);
321     }
322     public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
        IOException {
323         response.setContentType("text/html; charset=utf-8");
324         PrintWriter out = response.getWriter();
325         // 입력 파라미터 얻기
326         String product = request.getParameter("product");
327         //세션객체 얻기
328         HttpSession session = request.getSession();
329         ArrayList<String> list = (ArrayList<String>)session.getAttribute("product");
330         if( list == null ){
331             list = new ArrayList<String>();
332             list.add(product);
333             session.setAttribute("product", list);
334         }else{
335             list.add(product);
336         }
337         out.print("<html><body>");
338         out.print("Product 추가");
339         out.print("<a href='CartBasket'>장바구니 보기</a>");
340         out.print("</body></html>");
341     }
342 }

```

### 3)CartBasketServlet.java

```

346 public class CartBasketServlet extends HttpServlet {
347     public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
        IOException {
348         doPost(request, response);
349     }
350     public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
        IOException {
351         response.setContentType("text/html; charset=utf-8");
352         PrintWriter out = response.getWriter();
353         out.print("<html><body>");
354         out.print("장바구니 리스트");
355         //세션객체 얻기

```

```

356         HttpSession session = request.getSession(false);
357         session.setMaxInactiveInterval(200);
358         if( session != null ){
359             ArrayList<String> list = (ArrayList<String>)session.getAttribute("product");
360             out.print("상품: " + list + "<br>");
361         }else{
362             out.print("세션 없음" + "<br>");
363         }
364         out.print("<a href='product.html'>상품 선택 페이지</a><br>");
365         out.print("<a href='CartDelete'>장바구니 비우기</a><br>");
366         out.print("</body></html>");
367     }
368 }
369

```

#### 4)CartDeleteServlet.java

```

371
372 public class CartDeleteServlet extends HttpServlet {
373     public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException,
374     IOException {
375         doPost(request, response);
376     }
377     public void doPost(HttpServletRequest request, HttpServletResponse response) throws ServletException,
378     IOException {
379         response.setContentType("text/html; charset=utf-8");
380         PrintWriter out = response.getWriter();
381         out.print("<html><body>");
382         out.print("장바구니 비웠음!!");
383         //세션객체 얻기
384         HttpSession session = request.getSession(false);
385         if( session != null ){
386             session.invalidate();
387         }else{
388             out.print("세션 없음" + "<br>");
389         }
390         out.print("<a href='product.html'>상품 선택 페이지</a><br>");
391         out.print("</body></html>");
392     }
393 }

```