

## JSP Servlet 기초(8) 세션

JSP 2018. 1. 29. 18:39

이번엔 jsp 세션에 대해 알아 보겠습니다  
앞에서 웹브라우저와의 관계를 유지하는 수단으로 쿠키를 알아보았습니다.  
세션도 쿠키와 마찬가지로 서버와의 관계를 유지하기 위한 수단입니다  
하지만 쿠키와 달리 클라이언트에 값을 저장하는것이 아니라  
서버상에 객체로 존재 합니다. 세션은 서버에서만 접근이 가능합니다.  
그래서 쿠키보다 보안이 좋고 저장할수 있는 데이터에 한계가 없습니다.

세션을 알아보기전 간단한 세션 관련 메소드를 알아 보겠습니다.

`setAttribute()` : 세션에 데이터를 저장합니다.  
`getAttribute()` : 세션에서 데이터를 얻습니다.  
`getAttributeNames()` : 세션에 저장되어 있는 모든 데이터의  
이름(유니크한 키값)을 얻습니다.  
`getId()` : 세션의 유니크한 아이디를 얻습니다  
`isNew()` : 세션이 최초 생성되었는지 이전에 생성된 세션인지를 구분합니다.  
`getMaxInactiveInterval()` : 세션의 유효시간을 얻습니다  
가장 최근 요청 시점을 기준으로 카운트 됩니다.  
`removeAttribute()` : 세션에서 특정 데이터를 제거합니다.  
`invalidate()` : 세션의 모든 데이터를 삭제합니다.

이제 세션실습에 필요한 파일을 생성합니다.  
세션을 생성할 `sessionSet.jsp` 와 세션을 확인할 `sessionGet.jsp` 를 만들어줍니다.

`sessionSet.jsp` 를 먼저 보면

```
1 <body>
2
3     <%
4         session.setAttribute("SessionName", "SessionValue");
5         session.setAttribute("asdfg", 12345);
6     %>
7
8     <a href="sessionGet.jsp">session Get</a>
9
10 </body>
11
```

Colored by Color Scripter cs

`session.setAttribute`로 세션을 set(정의) 해줍니다.  
세션의 이름은 `SessionName` , 그 안에 들어갈 값(value)는  
`SessionValue` 로 설정합니다  
그밑에 세션은 이름을 `asdfg`, 값(value)은 `12345` 로 설정한다.  
이렇게 총 2개의 세션을 생성한것이다.

다음 하이퍼 링크로 session Get 을 눌러 sessionGet.jsp 로 이동되게 합니다.

`sessionGet.jsp` 는 아래와 같이 입력합니다.

```

1 <body>
2
3 <%
4     Object obj1 = session.getAttribute("SessionName");
5     String SessionName = (String)obj1;
6     out.println(SessionName + "<br />");
7
8     Object obj2 = session.getAttribute("asdfg");
9     Integer asdfg = (Integer)obj2;
10    out.println(asdfg + "<br />");
11
12    out.println("***** <br />");
13
14    String sName;
15    String sValue;
16    Enumeration enumeration = session.getAttributeNames();
17    while(enumeration.hasMoreElements()){
18        sName = enumeration.nextElement().toString();
19        sValue = session.getAttribute(sName).toString();
20        out.println("sName : " + sName + "<br />");
21        out.println("sValue : " + sValue + "<br />");
22    }
23
24    out.println("***** <br />");
25
26    String sessionID = session.getId();
27    out.println("sessionID : " + sessionID + "<br />");
28    int sessionInter = session.getMaxInactiveInterval();
29    out.println("sessionInter : " + sessionInter + "<br />");
30
31    out.println("***** <br />");
32
33    session.removeAttribute("SessionName");
34    Enumeration enumeration1 = session.getAttributeNames();
35    while(enumeration1.hasMoreElements()){
36        sName = enumeration1.nextElement().toString();
37        sValue = session.getAttribute(sName).toString();
38        out.println("sName : " + sName + "<br />");
39        out.println("sValue : " + sValue + "<br />");
40    }
41
42    out.println("***** <br />");
43
44    session.invalidate();
45    if(request.isRequestedSessionIdValid()) {
46        out.println("session valid");
47    } else {
48        out.println("session invalid");
49    }
50 %>
51
52 </body>
53

```

Colored by Color Scripter cs

먼저 4line에서 Object obj1 = session.getAttribute("SessionName");  
을 보면 sessionSet.jsp 에서 SessionName 이라는 이름으로 만든 세션을

session.getAttribute 으로 받아온다. 받아온 데이터를 Object 형으로 obj1 이라는 이름으로 다시 저장하는데

전에 request.getParameter 로 파라미터값을 요청 받아오면 String 으로 반환되어 값을 저장 할때 String 형으로 저장 했다 이와 마찬가지로 session.getAttribute 를 이용하여 세션값을 가져오면 Object 형으로 값이 반환되기 때문에 Object 형으로 값을 저장하는 것이다.

다음 obj1에 저장된 세션값을 다시 String 형으로 형변환 시킨후 SessionName 라는 String 형식 의 값으로 저장한다.  
String SessionName = (String)obj1;

out.println(SessionName + "<br />");을 통해 SessionName 의 값을 출력한다  
출력값은 SessionValue 가 될것이다.

8,9,10 line 은 두번째 세션을 똑같이 정의한다.

다음 16line Enumeration enumeration = session.getAttributeNames();  
에서 Enumeration 은 열거형 인터페이스로 저장된 값들을 직렬화로 저장한다.

#### \* Enumeration 인터페이스 메소드

Enumeration 메소드	설명
boolean hasMoreElements()	Vector로 부터 생성된 Enumeration의 요소가 있으면 true, 아니면 false 반환
Object nextElement()	Enumeration 내의 다음 요소를 반환한다.

즉 session.getAttributeNames 로 세션의 이름들을 불러와 enumeration에 저장한다음 반복문인 while 문으로 세션의 이름을 뽑아 내는거다.  
while문 안의enumeration.hasMoreElements()는 요소가 존재 하면 true를 반환 한다 결국 while(true) 가 되기 때문에 반복실행 된다.  
다음 sName 과 sVale 에 세션에 저장된 이름과 값을 저장한다.

```
sName = enumeration.nextElement().toString();  
sValue = session.getAttribute(sName).toString();
```

enumeration.nextElement()을 사용하여 enumeration내의 요소(세션이름)를 toString 으로 문자열로 변환하여 sName 에 저장한다.

다음 sValue 에서는 sName 에 저장된 세션의 이름으로부터 해당 세션의 값을 가져와서 toString 으로 문자열 변환후 sValue 에 저장한다.

다음 sName 과 sValue 를

```
out.println("sName : " + sName + "<br />");  
out.println("sValue : " + sValue + "<br />");
```

출력한다

```
다음 String sessionID = session.getId();  
out.println("sessionID : " + sessionID + "<br />");
```

getId()를 통해 세션의 유니크한 ID 값을 얻어와 출력한다

```
int sessionInter = session.getMaxInactiveInterval();
```

```
out.println("sessionInter : " + sessionInter + "<br />");
```

에서는 세션의 유효기간을 얻어와 출력한다 세션의 유효기간은  
따로 설정 하지않는한 서버에 기본적으로 저장된**1800(30분)** 으로 출력된다.

33line에서는 `SessionName` 라는이름을 가진 세션을 삭제하는것이다.  
그후 열거형인터페이스로 다시 세션들의 이름들을 불러와  
출력한다. `SessionName` 이름의 세션이 삭제 되었으므로  
출력값은 `asdfg` 의이름을 가진 세션값만 출력될것이다.

다음 44line에서는 `session.invalidate()`; 메소드로 모든 세션을 삭제하기  
때문에 출력문에 아무런 세션도 출력되지 못할것이다.

`sessionSet.jsp` 를 실행시켜보면 결과값은

```
SessionValue
12345
*****
sName : SessionName
sValue : SessionValue
sName : asdfg
sValue : 12345
*****
sessionID : 35374177C49104D06C537BF4A54B86FB
sessionInter : 1800
*****
sName : asdfg
sValue : 12345
*****
session invalid
```

위와같이 출력될것이다.