

24.02.06[JSP]

🕒 작성 일시	@2024년 2월 6일 오전 10:02
☰ 태그	

데이터 저장 내장 객체

pageContext: **한 페이지 만**

request: 내장 객체는 **한 번의 요청**을 처리하는 과정 안에서 유효

session: 기본 세션 타임 아웃 30분으로 설정(**브라우저 종료**)

application : 서버 종료 시 유효 범위 종료(**톰캣 종료**)

```
pageContext.setAttribute("id", id); //한 페이지 만
```

```
request.setAttribute("id", id); //한번에 요청 처리
```

```
session.setAttribute("id", id); //기본 세션 타임 아웃 30분으로 설정(브라우저 종료)
```

```
application.setAttribute("id", id); //서버 종료시 유효 범위 종료(톰캣 종료)
```

게시판

#게시글 리스트#

no	제목	작성자	작성일	조회수
20	감사합니다.	회원1	2016-04-27 05:44:32.0	162
19	궁금한게 해결 되었네요	회원8	2016-04-27 05:50:21.0	77
18	저두 궁금했는데	회원7	2016-04-27 05:44:32.0	81
17	아 줄 바꿈 문제 해결 했습니다.	관리자	2016-04-27 04:58:45.0	35
16	감사합니다.	관리자	2016-04-27 03:40:43.0	16
15	그러게요	회원3	2016-04-27 04:59:15.0	46
14	회원이면 당연한 것을...	회원3	2016-04-27 05:44:32.0	38
13	별 말씀을 다하시네요	회원1	2016-04-27 05:44:32.0	79
12	당연하죠~	회원6	2016-04-27 05:44:32.0	9
11	저두요~	회원5	2016-04-27 05:44:32.0	15
10	내용을 줄 바꿈하지 않아서...	관리자	2016-04-27 05:04:23.0	64
9	네 잘 알겠습니다.	회원4	2016-04-27 03:37:44.0	97
8	정말 정말 감사합니다.	관리자	2016-04-27 05:44:32.0	132
7	사이트 발전에 관심이 많은 사람입니다.	회원3	2016-04-27 05:03:52.0	82
6	많은 회원님들께서 관심을 가져주셔서 감사합니다.	관리자	2016-04-27 16:30:41.0	36
5	당연히 회원이면 관심을 가져야죠^_^	회원1	2016-04-27 05:44:32.0	42
4	저두 궁금했었는데	회원2	2016-04-27 05:44:32.0	103
3	관심을 가져 주셔서...	관리자	2016-04-27 05:44:32.0	35
2	공지 사항 잘 읽었습니다.	회원1	2016-04-27 05:43:38.0	53
1	공지 사항 입니다.	관리자	2016-04-27 01:44:58.0	15

게시판[controller]

```

1 package com.jspstudy.ch6.controller;
2
3*import java.io.IOException;
16
17 @WebServlet("/boardList01")
18 public class BoardListController01 extends HttpServlet {
19     private static final long serialVersionUID = 1L;
20
21     protected void doGet(HttpServletRequest request, HttpServletResponse response)
22         throws ServletException, IOException {
23
24         // BoardDao01 객체를 생성하고 DBb에서 게시물 리스트를 읽어온다
25         BoardDao01 boardDao = new BoardDao01();
26         ArrayList<Board> bList = boardDao.boardList();
27
28
29         //요청 처리 결과 - 모델(Model)을 HttpServletRequest 객체의 속성에 저장
30         request.setAttribute("bList", bList);
31
32         //View 페이지로 제어를 이동해 모델을 화면에 출력 - 포워딩
33         // /WEB-INF/board/boardList01.jsp
34         RequestDispatcher rd =
35             request.getRequestDispatcher("/WEB-INF/board/boardList01.jsp");
36         rd.forward(request, response);
37     }
38
39     String user = "hr";
40     String pass = "hr";
41     String driver = "oracle.jdbc.driver.OracleDriver";
42     String url = "jdbc:oracle:thin:@localhost:1521:xe";
43     Connection conn = null;
44     PreparedStatement pstmt = null;
45     ResultSet rs = null;
46     ArrayList<Board> bList= null;
47
48
49 }

```

게시판[model]

```

package com.jspstudy.ch06.dao;

import java.sql.Connection;

public class BoardDao01 {
    private static final String USER = "hr";
    private static final String PASS = "hr";
    private static final String DRIVER = "oracle.jdbc.driver.OracleDriver";
    private static final String URL = "jdbc:oracle:thin:@localhost:1521:xe";

    private Connection conn = null;
    private PreparedStatement pstmt = null;
    private ResultSet rs = null;

    public BoardDao01() {
        try {
            Class.forName(DRIVER);
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        }
    }

    //DB에서 게시판 리스트를 읽어와 반환하는 메서드
    public ArrayList<Board> boardList() {
        ArrayList<Board> bList = null;
        String select = "SELECT * FROM jsppbs ORDER BY no DESC";
        try {
            // 1. 접속하려는 DBMS의 접속 드라이버 로딩
            Class.forName(DRIVER);

            // 2. 데이터베이스 연결 -
            conn = DriverManager.getConnection(URL, USER, PASS);

            // 3. 활성화된 Connection 객체로부터 쿼리를 발행해 주는 PreparedStatement 객체를 구한다.
            pstmt = conn.prepareStatement(select);

            // 4. PreparedStatement 객체를 이용해 쿼리를 발행한다.
            // SELECT 쿼리는 executeQuery() => 쿼리 결과로 조회된 데이터의 집합 = ResultSet(가상의 테이블)
            // INSERT, UPDATE, DELETE executeUpdate() 메서드 => 반환 값은 테이블에서 수정된 행의 갯수를 정수로 반환
            rs = pstmt.executeQuery();

            // 게시판 글을 담은 리스트 객체 생성
            bList = new ArrayList<Board>();

            // 테이블 조회 결과를 반복문 안에서 Board 객체에 담고 다시 Board 객체를 ArrayList에 담는다.
            while (rs.next()) {
                Board b = new Board();
                b.setNo(rs.getInt(1));
                b.setTitle(rs.getString("title"));
                b.setWriter(rs.getString(3));
                b.setRegDate(rs.getTimestamp(5));
                b.setReadCount(rs.getInt("read_count"));

                bList.add(b);
            }
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            // 6. DB 작업에 사용했던 객체를 닫는다.
            // 앞에서 가져온 반대순으로 닫는다.
            try {
                if (rs != null) rs.close();
                if (pstmt != null) pstmt.close();
                if (conn != null) conn.close();
            } catch (SQLException e) {}
        }
        return bList;
    }
}

```

게시판 view

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import="java.sql.*, java.util.*, com.jspstudy.ch06.vo.*" %>
<%@ page import="com.jspstudy.ch06.vo.Board" %>
<%@ page import="java.util.ArrayList" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
<style type="text/css">
    table{
        border: 1px solid blue;
        border-collapse: collapse;
    }
    td {
        border: 1px dotted blue;
        padding: 5px 10px;
    }
</style>
</head>
<body>
<h1>#게시글 리스트#</h1>
<table>
<tr>
<th>no</th>
<th>제목</th>
<th>작성자</th>
<th>작성일</th>
<th>조회수</th>
</tr>

<%-- 게시글이 있는 경우 --%>
<c:if test="${not empty bList}">
<c:forEach var="board" items="${bList}">
<tr>
<td>${board.no}</td>
<td>${board.title}</td>
<td>${board.writer}</td>
<td>${board.regDate}</td>
<td>${board.readCount}</td>
</tr>
</c:forEach>
</c:if>
<%-- 게시글이 없는 경우 --%>
<c:if test="${empty bList}">
<tr>
<td colspan="5">게시글이 존재하지 않음</td>
</tr>
</c:if>

</table>
</body>
</html>

```

커먼아파치(<https://commons.apache.org/>)

DBCP

DBCP	Database connection pooling services.
------	---------------------------------------

pool

commons-pool2-2.6.2-bin.zip

logging

commons-logging-1.3.0-bin.zip

context

```
1 <?xml version="1.0" encoding="UTF-8"?>
  Bind to grammar/schema...
2 <Context>
3   <Resource name="jdbc/bbsDBPool"
4     auth="Container"
5     type="javax.sql.DataSource"
6     driverClassName="oracle.jdbc.driver.OracleDriver"
7     url="jdbc:oracle:thin:@localhost:1521:xe"
8     username="hr"
9     password="hr"
10    factory="org.apache.commons.dbcp2.BasicDataSourceFactory"
11    maxActive="10"
12    maxIdle="5" />
13 </Context>
14 <!--
15   톰캣에 기본 내장되어 있는 DBCP를 사용할 경우 DBCP Factory 클래스 지정
16   이클립스에 설치된 톰캣 서버의 context.xml 또는 server.xml에 Context를
17   설정하는 경우 factory 클래스는 아래와 같이 지정하면 된다.
18   -->
19 <!-- factory="org.apache.tomcat.dbcp.dbcp.BasicDataSourceFactory" -->
```