

JSP 게시판 강좌 따라하기 6강

회원가입 기능 구현하기

한 명의 회원정보를 입력하는 회원가입 폼에서 아이디, 비밀번호, 이름, 성별, 이메일을 입력해서 회원가입 버튼을 누르면 실제로 데이터베이스에 회원정보가 전송되는 과정입니다.

join 메소드

'UserDAO' 클래스에서 회원가입 기능을 담당하는 'join' 메소드를 추가합니다

```
public int join(User user) {
    String sql = "insert into user values(?, ?, ?, ?, ?)";
    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, user.getUserID());
        pstmt.setString(2, user.getUserPassword());
        pstmt.setString(3, user.getUserName());
        pstmt.setString(4, user.getUserGender());
        pstmt.setString(5, user.getUserEmail());
        return pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return -1;
}
```

```
public int join(User user) {
    String sql = "insert into user values(?, ?, ?, ?, ?)";
    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, user.getUserID());
        pstmt.setString(2, user.getUserPassword());
        pstmt.setString(3, user.getUserName());
        pstmt.setString(4, user.getUserGender());
        pstmt.setString(5, user.getUserEmail());
```

```

        return pstmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    }
    return -1;
}

```

public int join(User user) - 'User' 클래스 인스턴스를 생성하여 한 명의 회원정보를 담는다	
insert into user values(?, ?, ?, ?, ?);	'user' 테이블에 데이터를 입력하는 쿼리문
pstmt = conn.prepareStatement(sql);	'sql' 쿼리문을 대기시킨다
pstmt.setString(1, user.getUserID());	'1'번째 물음표에 실제 사용자가 입력한 'UserID'를 셋팅
pstmt.executeUpdate();	실제 DB에서 'insert'를 정상적으로 입력하면 'Query 1 OK' 라고 뜨는데 그 숫자를 받아온다. 따라서 정상적으로 데이터가 입력되었다면 반드시 1 이상의 숫자가 리턴될 것이다.

'joinAction.jsp'

```

5 <% request.setCharacterEncoding("utf-8"); %>
6 <jsp:useBean id="user" class="user.User" scope="page" />
7 <jsp:setProperty name="user" property="userID" />
8 <jsp:setProperty name="user" property="userPassword" />
9 <jsp:setProperty name="user" property="userName" />
10 <jsp:setProperty name="user" property="userGender" />
11 <jsp:setProperty name="user" property="userEmail" />
12 <!DOCTYPE html>

```

'loginAction.jsp' 페이지에서는 로그인 시 필요한 아이디와 비밀번호만 받아오면 됐었지만, 'joinAction.jsp' 페이지에서는 회원가입 페이지에서 받아오는 데이터를 다 받아와야 하기 때문에 회원 정보 요소를 모두 추가합니다.

```

3 <body>
4 <%
5
6     if(user.getUserID() == null || user.getUserPassword() == null ||
7       user.getUserGender() == null || user.getUserEmail() == null)
8       PrintWriter script = response.getWriter();
9       script.println("<script>");
10      script.println("alert('입력이 안 된 사항이 있습니다')");
11      script.println("history.back()");
12      script.println("</script>");
13    }else{
14      UserDao userDao = new UserDao();
15      int result = userDao.join(user);
16      if(result == -1){
17        PrintWriter script = response.getWriter();
18        script.println("<script>");
19        script.println("alert('이미 존재하는 아이디입니다')");
20        script.println("history.back()");
21        script.println("</script>");
22      }else {
23        PrintWriter script = response.getWriter();
24        script.println("<script>");
25        script.println("alert('회원가입 성공')");
26        script.println("location.href='main.jsp'");
27        script.println("</script>");
28      }
29    }
30  }
31  %>
32 </body>

```

```

<%@page import="java.io.PrintWriter"%>
<%@page import="user.UserDAO"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<% request.setCharacterEncoding("utf-8"); %>
<jsp:useBean id="user" class="user.User" scope="page" />
<jsp:setProperty name="user" property="userID" />
<jsp:setProperty name="user" property="userPassword" />
<jsp:setProperty name="user" property="userName" />
<jsp:setProperty name="user" property="userGender" />
<jsp:setProperty name="user" property="userEmail" />
<!DOCTYPE html>
<html>

```

```

<head>
<meta charset="UTF-8">
<title>JSP 게시판 웹 사이트</title>
</head>
<body>
    <%
        if(user.getUserID() == null || user.getUserPassword() == null || user.getUserName() ==
null ||

            user.getUserGender() == null || user.getUserEmail() == null){
                PrintWriter script = response.getWriter();
                script.println("<script>");
                script.println("alert('입력이 안 된 사항이 있습니다')");
                script.println("history.back()");
                script.println("</script>");
            }else{
                UserDao userDao = new UserDao();
                int result = userDao.join(user);
                if(result == -1){
                    PrintWriter script = response.getWriter();
                    script.println("<script>");
                    script.println("alert('이미 존재하는 아이디입니다')");
                    script.println("history.back()");
                    script.println("</script>");
                }else {
                    PrintWriter script = response.getWriter();
                    script.println("<script>");
                    script.println("alert('회원가입 성공')");
                    script.println("location.href='main.jsp'");
                    script.println("</script>");
                }
            }
    %>
</body>
</html>

```

실제 사용자가 입력한 회원 정보 중에서 데이터가 하나라도 누락된 것이 있으면 '입력이 안 된 사항이 있다' 라고 알림창을 띄워주고 다시 이전 화면으로 돌아가는 로직을 추가합니다.

만약 입력이 다 되었더라도, 아이디가 중복이 되면 다시 이전 페이지로 돌아가게 로직을 하나 추가합니다.우리가 테이블을 작성할 때 'UserID'는 'primarykey'로 설정했었기 때문에 중복되지 않는 하나만 존재할 수 있는 고유키가 됩니다. 실제 사용자가 입력한 정보가 이미 존재하는 아이디를 입력했다라면 DB에 회원정보가 정상적으로 입력이 되지 않아 '-1'이 반환될 것입니다.

모든 정보가 올바르게 입력 되었다면 '1' 이상이 반환될 것이고 '회원가입 성공'이라는 알림창이 뜨며 'main.jsp' 페이지로 이동할 것입니다.(아직까지 'main.jsp' 페이지는 작성하지 않았으니 '404에러' 가 뜰 것입니다)

회원가입 기능 테스트

JSP 게시판 웹 사이트 메인 게시판

1 접속하기 ▾

로그인
회원가입 2

로그인 화면

아이디

비밀번호

로그인

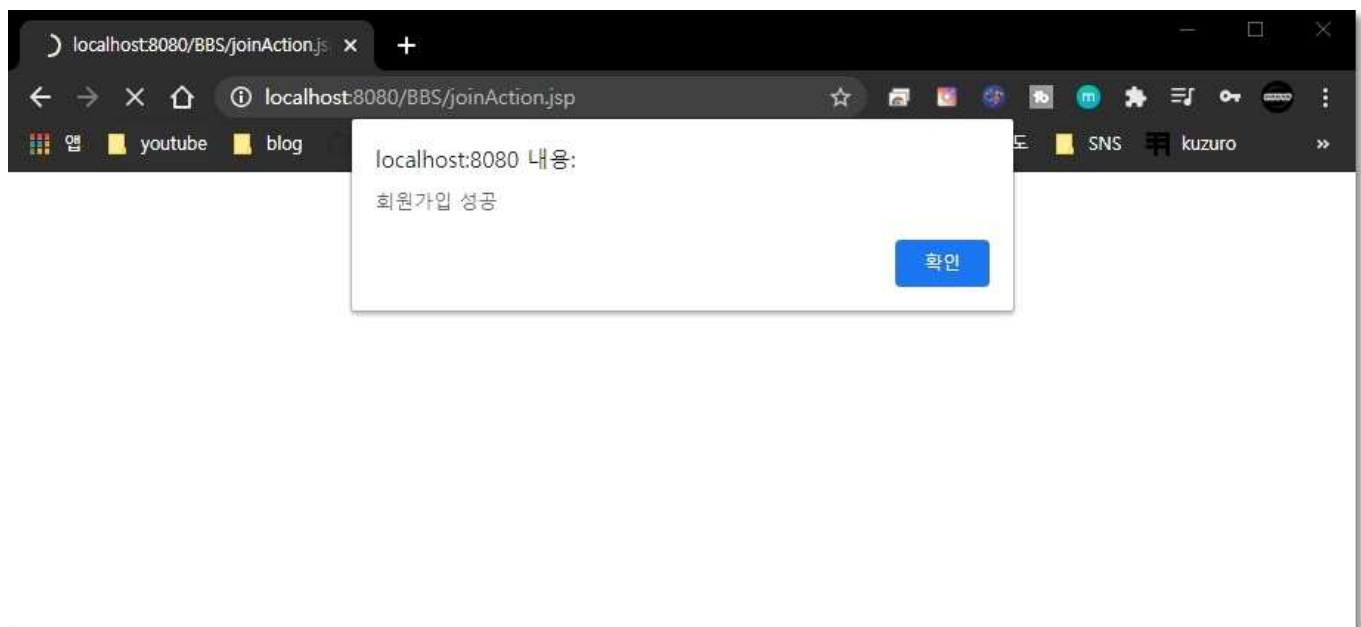
localhost:8080/BBS/join.jsp

[index.jsp 페이지 실행] - [접속하기] - [회원가입]

JSP 게시판 웹 사이트 메인 게시판 접속하기 ▾

회원가입 화면

항목에 맞게 데이터 입력 후 '회원가입' 버튼 누르기



데이터가 올바르게 입력되었을 때 설정한 알림창이 떴습니다
실제로 데이터가 'MariaDB' 에 입력이 되었는지 확인해보겠습니다

```
MySQL Client (MariaDB 10.2 (x64)) - "C:\Program Files\MariaDB 10.2\bin\mysql.exe" "--defaults-file=C:\Program Files\MariaDB 10.2\data#...
Enter password: *****
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 15
Server version: 10.2.32-MariaDB mariadb.org binary distribution
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use bbs;
Database changed
MariaDB [bbs]> select * from user;
+----+-----+-----+-----+-----+
| userID | userPassword | userName | userGender | userEmail |
+----+-----+-----+-----+-----+
| admin  | 0000         | admin   | 남자       | admin@admin.com |
| gildong | 123          | 홍길동  | 남자       | gildong@naver.com |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

MariaDB [bbs]>
```

방금 회원정보를 입력한 데이터가 실제로 DB 안에 저장된 것을 확인할 수 있습니다