

회원관리형 게시판(MVC2)



Jweb Project



회원 관리형 게시판 (JSP mvc2)

[Home](#)[회원 가입](#)[게시판](#)[로그인](#)

Welcome to World Friends



현재 접속 시각: 0:10:50 AM

© World Job Friends : JMP Korea

회원 관리형 게시판 프로젝트

프로그램 소스 목록 1 – 메인, 회원가입

파일 이름	역 할
index.jsp	초기 화면
main.jsp	메인 화면
menu.jsp	메뉴 파일
footer.jsp	푸터 파일
members.sql	회원과 게시판 테이블을 저장할 sql 파일
style.css	스타일을 디자인하는 css 파일
memberForm.jsp	회원 가입 폼 파일
checkMember.js	회원가입 폼에 여러가지 유효한지 체크하는 js파일
Member.java	회원 데이터 매핑을 위한 빈즈 클래스
MemberDAO.java	Member를 관리 사용하는 클래스
MainController	Member와 MemberDAO를 사용하여 model 자료를 저장하고 view에 전달하는 서비스를 담당하는 로직



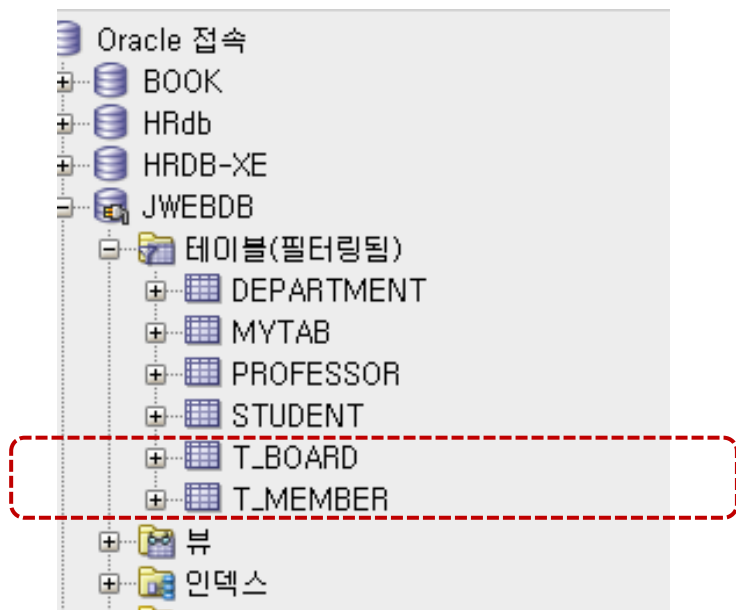
회원 관리형 게시판 프로젝트

데이터베이스 모델링(회원과 게시판 테이블 설계)



회원 관리형 게시판 프로젝트

JWEBDB 스키마에 t_member 테이블 만들기



```
-- member 테이블 생성 --  
CREATE TABLE t_member(  
    memberId    VARCHAR2(20),  
    passwd      VARCHAR2(20) NOT NULL,  
    name        VARCHAR2(30) NOT NULL,  
    gender      VARCHAR2(10),  
    joinDate    DATE DEFAULT SYSDATE,  
    PRIMARY KEY(memberId)  
);  
  
DROP TABLE t_member CASCADE CONSTRAINTS;  
  
SELECT * FROM t_member;
```

회원 관리형 게시판 프로젝트

JWEBDB 스키마에 t_board 테이블 만들기

```
-- t_board 테이블 만들기 --  
CREATE TABLE t_board(  
    bnum NUMBER(4) PRIMARY KEY,  
    title VARCHAR2(100) NOT NULL,  
    content VARCHAR2(2000) NOT NULL,  
    regDate DATE DEFAULT SYSDATE,  
    memberId VARCHAR2(20),  
    CONSTRAINT FK_MemberBoard FOREIGN KEY(memberId)  
    REFERENCES t_member(memberId)  
);
```

```
-- 시퀀스 생성 --  
CREATE SEQUENCE bSeq  
    INCREMENT BY 1  
    START WITH 1  
    MINVALUE 1  
    MAXVALUE 1000  
    NOCYCLE  
    NOCACHE;
```

```
-- 외래키 제약 조건 삭제  
ALTER TABLE t_board DROP CONSTRAINT FK_MemberBoard;
```

```
-- 외래키 제약 조건 설정 : t_member의 자료 삭제 가능  
ALTER TABLE t_board ADD CONSTRAINT FK_MemberBoard FOREIGN KEY(memberId)  
    REFERENCES t_member(memberId) ON DELETE CASCADE;
```

```
-- 조회수 칼럼 추가  
ALTER TABLE t_board ADD (hit NUMBER(5));
```



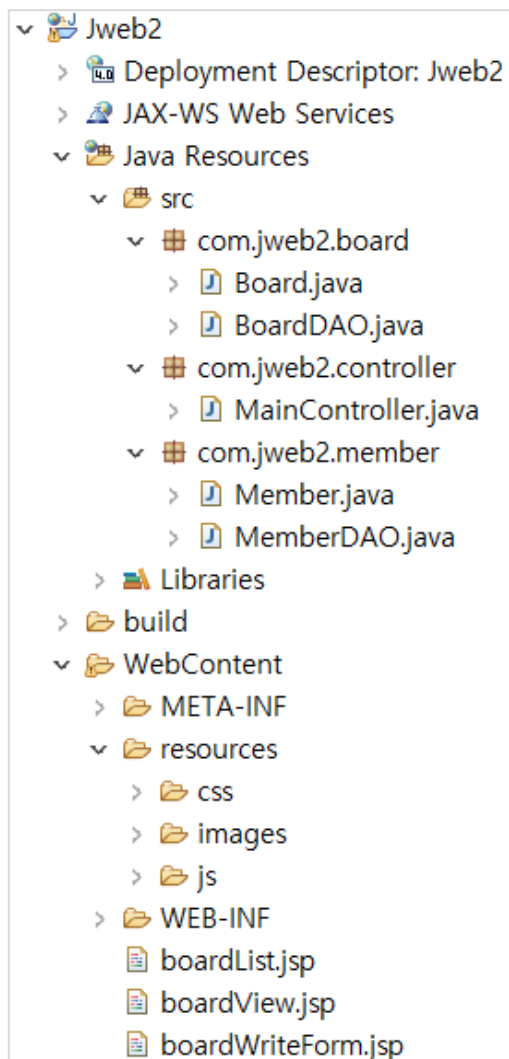
회원 관리형 게시판 프로젝트

index.jsp

```
<title>회원 관리</title>
</head>
<body>
    <script type="text/javascript">
        location.href = 'main.jsp';
    </script>
</body>
```

footer.jsp

```
<footer>
    <p>&copy; World Job Friends : JMP Korea</p>
</footer>
```



회원 관리형 게시판 프로젝트

```
<title>메뉴</title>
<link rel="stylesheet" href="resources/css/style.css" />
</head>
<body>
  <nav>
    <c:choose>
      <c:when test="${empty sessionId}">
        <ul>
          <li><a href="main.jsp">Home</a>
          <li><a href="memberForm.do">회원 가입</a>
          <li><a href="boardList.do">게시판</a>
          <li><a href="LoginForm.do">로그인</a>
        </ul>
      </c:when>
      <c:otherwise>
        <ul>
          <li><a href="main.jsp">Home</a>
          <li><a href="memberView.do">회원 수정</a>
          <li><a href="boardList.do">게시판</a>
          <li><a href="logout.do">[${name}님]로그아웃</a>
        </ul>
      </c:otherwise>
    </c:choose>
  </nav>
```

menu.jsp



회원 관리형 게시판 프로젝트

main.jsp

```
<jsp:include page="menu.jsp" />
<div id="container">
  <div class="title">
    <h1>Welcome to World Friends</h1>
  </div>
  <div>
    <p>
    <p>
    <%
      Date day = new java.util.Date();
      String am_pm;
      int hour = day.getHours();
      int minute = day.getMinutes();
      int second = day.getSeconds();
      if (hour / 12 == 0) {
        am_pm = "AM";
      } else {
        am_pm = "PM";
        hour = hour - 12;
      }
      String CT = hour + ":" + minute + ":" + second + " " + am_pm;
      out.println("현재 접속 시각: " + CT + "\n");
      response.setHeader("Refresh", 5);
    %>
  </div>
</div>
```

회원 관리형 게시판 프로젝트

회원가입 및 회원 회원 수정

회원 가입

아이디	<input type="text" value="id"/>
비밀번호	<input type="password" value="password"/>
비밀번호 확인	<input type="password" value="password confirm"/>
이름	<input type="text"/>
성별	<input type="radio"/> 남 <input type="radio"/> 여
<input type="button" value="등록"/> <input type="button" value="취소"/>	

회원 관리형 게시판 프로젝트

style.css

```
#container{
    width: 100%;
    margin: 0 auto;
    text-align: center;
}
nav{margin-top: 30px; text-align: right; margin-right: 30px;}
nav ul {list-style-type: none;}
nav ul li {display: inline; padding: 10px 15px;
    margin-right: 20px; border-top: 3px solid #222;}
nav ul li:hover {background-color: rgb(196, 196, 196);}
nav ul li a {color: black; text-decoration: none;}

.title{height: 130px; font-size: 25px; background: #eee;
    line-height: 130px; color: #333; margin-bottom: 20px}
.msg{height: 80px; font-size: 1em; background: pink;
    line-height: 80px; color: brown; text-align: center;}
footer{margin-left: 25%; padding: 15px;}
```



회원 관리형 게시판 프로젝트

```
table{width: 400px; margin: 0 auto;}
table, td{
    border: 1px solid #ccc;
    border-collapse: collapse;
    height: 30px;
    padding: 5px;
}
input[type="text"],[type="password"]{
    width: 200px;
    height: 20px;
}
a{color: #000; text-decoration:none}
textarea{width: 90%; height: 300px;
    border: 0.2px solid #eee; padding: 10px}
#table1{width:500px; margin:0 auto}
#table2{width: 700px;margin: 0 auto;}
#table2, td{height: 30px;}
#table2 .w_title{width: 630px;}
#table2 .thead{background: rgb(219, 239, 252)}
#table3{width: 700px; margin: 0 auto;}
#table3 td:nth-child(2n-1){width: 100px;}
#table3 td:nth-child(2n){text-align: left; text-indent: 15px;}
#table3 .w_title{width: 530px; border: 0.2px solid #eee;
    padding: 5px;}
.search-form{padding-left: 310px; margin-bottom: 10px;}
```

style.css



회원 관리형 게시판 프로젝트

Member.java

```
package com.jweb.member;

import java.io.Serializable;
import java.util.Date;

public class Member implements Serializable{
    private static final long serialVersionUID = 1L;

    private String memberId;
    private String passwd;
    private String name;
    private String gender;
    private Date joinDate;

    public String getMemberId() {
        return memberId;
    }

    public void setMemberId(String memberId) {
        this.memberId = memberId;
    }

    public String getPasswd() {
        return passwd;
    }

    public void setPasswd(String passwd) {
        this.passwd = passwd;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getGender() {
        return gender;
    }

    public void setGender(String gender) {
        this.gender = gender;
    }

    public Date getJoinDate() {
        return joinDate;
    }

    public void setJoinDate(Date joinDate) {
        this.joinDate = joinDate;
    }
}
```



회원 관리형 게시판 프로젝트

MemberDAO.java

```
package com.jweb.member;

import java.sql.Connection;
import java.sql.Date;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;

public class MemberDAO {

    private String driverClass = "oracle.jdbc.OracleDriver";
    private String url = "jdbc:oracle:thin:@localhost:1521:xe";
    private String username = "jweb";
    private String password = "4321";

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



회원 관리형 게시판 프로젝트

//DB 연결 메서드

```
private void connDB() {  
    try {  
        Class.forName(driverClass);  
        conn = DriverManager.getConnection(url, username, password);  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
}
```

//연결 종료 메서드

```
private void disconnect() {  
    if(pstmt != null) {  
        try {  
            pstmt.close();  
        } catch (SQLException e) {  
            e.printStackTrace();  
        }  
    }  
    if(conn != null) {  
        try {  
            conn.close();  
        } catch (SQLException e) {  
            e.printStackTrace();  
        }  
    }  
}
```

//연결 종료 메서드

```
private void disconnectRS() {  
    if(rs != null) {  
        try {  
            rs.close();  
        } catch (SQLException e) {  
            e.printStackTrace();  
        }  
    }  
    if(pstmt != null) {  
        try {  
            pstmt.close();  
        } catch (SQLException e) {  
            e.printStackTrace();  
        }  
    }  
    if(conn != null) {  
        try {  
            conn.close();  
        } catch (SQLException e) {  
            e.printStackTrace();  
        }  
    }  
}
```



회원 관리형 게시판 프로젝트

//회원 추가(삽입) 메서드

```
public void addMember(Member member) {
    connDB();
    String sql = "INSERT INTO t_member(memberId, passwd, name, gender) VALUES (?, ?, ?, ?)";
    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, member.getMemberId());
        pstmt.setString(2, member.getPasswd());
        pstmt.setString(3, member.getName());
        pstmt.setString(4, member.getGender());
        pstmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        disconnect();
    }
}
```



회원 관리형 게시판 프로젝트

```
<title>회원 가입 폼</title>
<link rel="stylesheet" href="resources/css/style.css" />
<script type="text/javascript" src="resources/js/checkMember.js"></script>
</head>
<body>
    <jsp:include page="menu.jsp" />
    <div id="container">
        <div class="title">
            <h1>회원 가입</h1>
        </div>
        <div>
            <p><form name="regForm" action="memberAdd.do" method="post">
                <table>
                    <tr>
                        <td>아이디</td>
                        <td><input type="text" name="memberId" placeholder="id"></td>
                    </tr>
                    <tr>
                        <td>비밀번호</td>
                        <td><input type="password" name="passwd" placeholder="password"></td>
                    </tr>
                </table>
            </p>
        </div>
    </div>
</body>
```

memberForm.jsp



회원 관리형 게시판 프로젝트

```
<tr>
  <td>비밀번호 확인</td>
  <td><input type="password" name="passwd_confirm" placeholder="password confirm"></td>
</tr>
<tr>
  <td>이 름</td>
  <td><input type="text" name="name" required></td>
</tr>
<tr>
  <td>성 별</td>
  <td><input type="radio" name="gender" value="남">남
    <input type="radio" name="gender" value="여">여
  </td>
</tr>
<tr>
  <td colspan="2">
    <input type="button" value="등록" onclick="checkMember()">
    <input type="reset" value="취소">
  </td>
</tr>
```

memberForm.jsp

```
    </table>
  </form>
</div>
</div>
<jsp:include page="footer.jsp" />
```



회원 관리형 게시판 프로젝트

```
/*
    회원 가입 폼 유효성 검사
*/
function checkMember(){
    var form = document.regForm;
    var id = form.memberId.value;
    var pw1 = form.passwd.value;
    var pw2 = form.passwd_confirm.value;
    var name = form.name.value;

    var regExpW = /^[a-zA-Z0-9]{8,20}$/;
    var chk_num = pw1.search(/[0-9]/g);
    var chk_eng = pw1.search(/[a-zA-Z]/g);

    if(id.length < 4 || id.length > 12){
        alert("아이디는 4~12자까지 가능합니다.");
        form.memberId.select();
        return false;
    }
}
```

check.js



회원 관리형 게시판 프로젝트

```
if(!regExpPw.test(pw1) || chk_num < 0 || chk_eng < 0){  
    alert('비밀번호는 숫자와 영문자 조합으로 8~20자까지 가능합니다.');
```



```
    form.passwd.select();  
    return false;  
}  
  
if(pw1 != pw2){  
    alert("비밀번호를 동일하게 입력해 주세요.");  
    form.passwd_confirm.select();  
    return false;  
}  
  
if(name==""){  
    alert("이름을 입력해 주세요");  
    form.name.focus();  
    return false;  
}  
  
form.submit();  
}
```

check.js



회원 관리형 게시판 프로젝트

프로그램 소스 목록 2 – 회원보기 - 수정, 탈퇴

파일 이름	역 할
memberView.jsp	MemberDAO를 이용해 데이터 상세 보기 및 수정, 탈퇴 화면
loginForm.jsp	아이디와 비밀번호를 입력하여 로그인을 할 수 있는 화면
memberResult.jsp	로그인, 회원가입, 수정, 삭제 시 메시지를 출력하는 화면



회원 관리형 게시판 프로젝트

회원 정보, 로그인, 로그아웃

회원 수정

아이디	<input type="text" value="mount"/>
비밀번호	<input type="text" value="m123456"/>
비밀번호 확인	<input type="text" value="password confirm"/>
이름	<input type="text" value="백두산"/>
성별	<input checked="" type="radio"/> 남 <input type="radio"/> 여
가입일	<input type="text" value="2021-03-14"/>
<input type="button" value="수정"/> <input type="button" value="탈퇴"/>	

로그인

아이디	<input type="text"/>
비밀번호	<input type="password"/>
<input type="button" value="로그인"/> <input type="button" value="취소"/>	

Home

회원 수정

게시판

[백두산님]로그아웃

회원 관리형 게시판 프로젝트

회원 관리에 변동이 있을때 출력되는 화면

회원 정보

박마늘님 환영합니다.

회원 정보

회원 가입을 축하합니다. 로그인 해주세요

회원 정보

회원 정보가 수정되었습니다.

회원 정보

회원 정보가 삭제되었습니다.

회원 관리형 게시판 프로젝트

MemberDAO에서 추가 사항 – 회원 보기

```
//회원 보기
public Member getOneDB(String memId) {
    connDB();
    String sql = "SELECT * FROM t_member WHERE memberId = ?";
    Member member = new Member();
    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, memId);
        //쿼리 실행
        rs = pstmt.executeQuery();
        rs.next();
        member.setMemberId(rs.getString("memberId"));
        member.setPasswd(rs.getString("passwd"));
        member.setName(rs.getString("name"));
        member.setGender(rs.getString("gender"));
        member.setJoinDate(rs.getDate("joinDate"));
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {disconnectRS();}
    return member;
}
```



회원 관리형 게시판 프로젝트

MemberDAO에서 추가 사항 – 삭제하기

```
//특정 회원 삭제
public boolean delete(String memId) {
    connDB();
    String sql = "DELETE FROM t_member WHERE memberId = ?";
    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, memId);
        pstmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
        return false;
    } finally {
        disconnect();
    }
    return true;
}
```



회원 관리형 게시판 프로젝트

MemberDAO에서 추가 사항 - 수정하기

```
//수정, 변경처리
public void update(Member member) {
    connDB();
    String sql = "UPDATE t_member SET passwd=?, name=?, gender=? WHERE memberId=?";
    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, member.getPasswd());
        pstmt.setString(2, member.getName());
        pstmt.setString(3, member.getGender());
        pstmt.setString(4, member.getMemberId());
        pstmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        disconnect();
    }
}
```



회원 관리형 게시판 프로젝트

MemberDAO에서 추가 사항 - 로그인 체크

```
//로그인 체크
public int login(String memberId, String pwd) {
    connDB();
    String sql = "SELECT memberId, passwd FROM t_member WHERE memberId = ?";
    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, memberId);
        rs = pstmt.executeQuery();
        if(rs.next()) {
            String dbPw = rs.getString("passwd");
            if(dbPw.equals(pwd))
                return 1; //비밀번호 일치
            else
                return 0; //비밀번호 불일치
        }else {
            return -1; //아이디 불일치
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        disconnect();
    }
    return -2; //데이터베이스 오류
}
```



회원 관리형 게시판 프로젝트

MemberDAO에서 추가 사항 – 로그인한 memberId의 이름 가져오기

```
// 인증된 memberId의 이름 가져오기
public String getLoginNameById(String memberId) {
    connDB();
    String sql = "SELECT * FROM t_member WHERE memberId = ?";
    String name = null;
    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, memberId);
        rs = pstmt.executeQuery();
        if(rs.next()) {
            name = rs.getString("name");
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        disconnect();
    }
    return name;
}
```



회원 관리형 게시판 프로젝트

```
<title>회원 수정</title>
<link rel="stylesheet" href="resources/css/style.css" />
</head>
<script type="text/javascript">
    function checkForm(){
        var form = document.updateForm;
        var pw1 = form.passwd.value;
        var pw2 = form.passwd_confirm.value;

        if(pw1 != pw2){
            alert("비밀번호를 동일하게 입력해 주세요");
            return false;
        }
        form.submit();
    }
</script>
<body>
    <jsp:include page="menu.jsp" />
    <div id="container">
        <div class="title">
            <h1>회원 수정</h1>
        </div>
```

memberView.jsp



회원 관리형 게시판 프로젝트

```
<div>
  <form name="updateForm" action="memberUpdate.do?memberId=${member.memberId}"
    method="post">
    <p><table>
      <tr>
        <td>아이디</td>
        <td><input type="text" name="memberId" value="${member.memberId}" disabled></td>
      </tr>
      <tr>
        <td>비밀번호</td>
        <td><input type="text" name="passwd" value="${member.passwd}"></td>
      </tr>
      <tr>
        <td>비밀번호 확인</td>
        <td><input type="text" name="passwd_confirm" placeholder="password confirm"></td>
      </tr>
      <tr>
        <td>이 름</td>
        <td><input type="text" name="name" value="${member.name}"></td>
      </tr>
    </p>
  </form>
</div>
```

memberView.jsp



회원 관리형 게시판 프로젝트

memberView.jsp

```
<td>성 별</td>
<td>
<c:if test="${member.gender=='남'}">
    <input type="radio" name="gender" value="남" checked>남
    <input type="radio" name="gender" value="여" >여
</c:if>
<c:if test="${member.gender=='여'}">
    <input type="radio" name="gender" value="남">남
    <input type="radio" name="gender" value="여" checked>여
</c:if>
</td>
</tr>
<tr>
<td>가입일</td>
<td><input type="text" name="joinDate" value="${member.joinDate}" disabled></td>
</tr>
<tr>
<td colspan="2">
    <input type="button" value="수정" onclick="checkForm()">
    <a onclick="return confirm('정말로 탈퇴하시겠습니까?')"
        href="memberDelete.do?memberId=${member.memberId}">
        <input type="button" value="탈퇴">
    </a>
</td>
```

회원 관리형 게시판 프로젝트

```
<jsp:include page="menu.jsp" />
<div id="container">
  <div class="title">
    <h1>로그인</h1>
  </div>
  <div>
    <form action="loginProcess.do" method="post">
      <table>
        <tr>
          <td>아이디</td>
          <td><input type="text" name="memberId"></td>
        </tr>
        <tr>
          <td>비밀번호</td>
          <td><input type="password" name="passwd"></td>
        </tr>
        <tr>
          <td colspan="2">
            <input type="submit" value="로그인">
            <input type="reset" value="취소">
          </td>
        </tr>
      </table>
    </form>
  </div>
</div>
```

loginForm.jsp



회원 관리형 게시판 프로젝트

```
<body>
  <jsp:include page="menu.jsp" />
  <div id = "container">
    <div class="title"><h1>회원 정보</h1></div>
    <div class="msg">
      <c:choose>
        <c:when test="${loginResult==0 }">
          <script type="text/javascript">
            alert('아이디가 일치하지 않습니다. ');
            history.go(-1);
          </script>
        </c:when>
        <c:when test="${loginResult==-1}">
          <script type="text/javascript">
            alert('비밀번호가 일치하지 않습니다. ');
            history.go(-1);
          </script>
        </c:when>
        <c:when test="${loginResult==-2}">
          <script type="text/javascript">
            alert('데이터베이스 오류입니다. ');
            history.go(-1);
          </script>
        </c:when>
      </c:choose>
    </div>
  </div>
```

memberResult.jsp



회원 관리형 게시판 프로젝트

```
</c:choose>
<c:choose>
  <c:when test="${msg eq 'login' }">
    <h1>${name}님 환영합니다.</h1>
  </c:when>
  <c:when test="${msg eq 'join' }">
    <h1>회원 가입을 축하합니다.</h1>
  </c:when>
  <c:when test="${msg eq 'update' }">
    <h1>회원 정보가 수정되었습니다.</h1>
  </c:when>
  <c:when test="${msg eq 'delete' }">
    <h1>회원 정보가 삭제되었습니다.</h1>
  </c:when>
</c:choose>
</div>
<jsp:include page="footer.jsp" />
```

memberResult.jsp



회원 관리형 게시판 프로젝트

프로그램 소스 목록 게시판 - 글쓰기, 목록

파일 이름	역 할
boardWriteForm.jsp	게시판 글 쓰기 양식
checkBoard.js	글 쓰기 폼에 입력이 안된 항목이 있을 때 체크 js파일
BoardList.jsp	BoardDAO 객체를 통해 저장된 데이터를 가져와서 화면에 출력
boardView.jsp	게시글 1개 상세 보기 수정과 삭제 버튼으로 처리하는 화면
Board.java	게시판 데이터 매핑을 위한 빈즈 클래스
BoardDAO.java	Board를 관리 사용하는 클래스



회원 관리형 게시판 프로젝트

글쓰기 양식, 게시글 목록 화면

글쓰기 양식

글 제목

글 내용

글쓰기

취소

© World Job Friends

게시글 목록

번호	글제목	글쓴이	작성일	조회수
4	3번글 삭제 테스트	river	2021-03-18	1
2	밤이 깊어갑니다.	mount	2021-03-18	0
1	테스트 글 제목입니다.	mount	2021-03-18	1

글쓰기

회원 관리형 게시판 프로젝트

Board.java

```
public class Board implements Serializable{

    private static final long serialVersionUID = 1L;

    private int bnum;
    private String title;
    private String content;
    private Date regDate;
    private int hit;
    private String memberId;

    public int getBnum() {
        return bnum;
    }
    public void setBnum(int bnum) {
        this.bnum = bnum;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getContent() {
        return content;
    }
    public void setContent(String content) {
        this.content = content;
    }
    public Date getRegDate() {
        return regDate;
    }
    public void setRegDate(Date regDate) {
        this.regDate = regDate;
    }
    public int getHit() {
        return hit;
    }
    public void setHit(int hit) {
        this.hit = hit;
    }
    public String getMemberId() {
        return memberId;
    }
    public void setMemberId(String memberId) {
        this.memberId = memberId;
    }
}
```



회원 관리형 게시판 프로젝트

BoardDAO.java

```
public class BoardDAO {  
    private String driverClass = "oracle.jdbc.OracleDriver";  
    private String url = "jdbc:oracle:thin:@localhost:1521:xe";  
    private String username = "jweb";  
    private String password = "4321";  
  
    private Connection conn = null;  
    private PreparedStatement pstmt = null;  
    private ResultSet rs = null;  
  
    private void connDB() {  
        try {  
            Class.forName(driverClass);  
            conn = DriverManager.getConnection(url, username, password);  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```



회원 관리형 게시판 프로젝트

BoardDAO.java

```
//db연결 종료 - pstmt, conn
private void disconnect() {
    if(pstmt != null) {
        try {
            pstmt.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    if(conn != null) {
        try {
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

```
//db연결 종료 - rs, pstmt, conn
private void disconnectRS() {
    if(rs != null) {
        try {
            rs.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    if(pstmt != null) {
        try {
            pstmt.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
    if(conn != null) {
        try {
            conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```



회원 관리형 게시판 프로젝트

BoardDAO.java - 새글 추가

```
//새글 추가
public void insertBoard(Board board) {
    connDB();
    String sql = "INSERT INTO t_board (bnum, title, content, memberId)"
        + " VALUES (bSeq.NEXTVAL, ?, ?, ?)";

    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, board.getTitle());
        pstmt.setString(2, board.getContent());
        pstmt.setString(3, board.getMemberId());
        pstmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        disconnect();
    }
}
```



회원 관리형 게시판 프로젝트

BoardDAO.java

- 목록 조회

```
public ArrayList<Board> getListAll(){
    connDB();
    ArrayList<Board> list = new ArrayList<>();
    String sql = "SELECT * FROM t_board ORDER BY bnum DESC";
    try {
        pstmt = conn.prepareStatement(sql);
        rs = pstmt.executeQuery();
        while(rs.next()) {
            int bnum = rs.getInt("bnum");
            String title = rs.getString("title");
            String content = rs.getString("content");
            Date regDate = rs.getDate("regDate");
            int hit = rs.getInt("hit");
            String memberId = rs.getString("memberId");

            Board board = new Board(); //board 객체 생성
            board.setBnum(bnum);
            board.setTitle(title);
            board.setContent(content);
            board.setRegDate(regDate);
            board.setHit(hit);
            board.setMemberId(memberId);
            list.add(board); //list에 저장
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {disconnectRS();}
    return list;
}
```



회원 관리형 게시판 프로젝트

```
<title>글쓰기 창</title>
<link rel="stylesheet" href="resources/css/style.css" />
<script type="text/javascript" src="resources/js/checkBoard.js"></script>
</head>
<body>
    <c:if test="${empty sessionId}">
        <script type="text/javascript">
            alert('로그인이 필요합니다.');
            location.href='loginForm.jsp';
        </script>
    </c:if>
    <jsp:include page="menu.jsp" />
    <div id="container">
        <div class="title">
            <h1>글쓰기 양식</h1>
        </div>
```

boardWriteForm.jsp



회원 관리형 게시판 프로젝트

boardWriteForm.jsp

```
<div>
  <p><form name="writeForm" action="boardWriteAdd.do" method="post">
    <table id="table2">
      <tr>
        <td><input type="text" name="title" class="w_title" placeholder="글 제목"></td>
      </tr>
      <tr>
        <td><textarea name="content" placeholder="글 내용"></textarea></td>
      </tr>
      <tr>
        <td colspan="2">
          <input type="button" value="글쓰기" onclick="checkBoard()">
          <input type="reset" value="취소">
        </td>
      </tr>
    </table>
  </form>
</div>
```



회원 관리형 게시판 프로젝트

```
/* 글쓰기 양식 유효성 검사
   글제목과 글내용이 비어있는 경우
*/
function checkBoard(){
    var form = document.writeForm;
    var title = form.title.value;
    var content = form.content.value;

    if(title==""){
        alert("제목을 입력해 주세요");
        form.title.focus();
        return false;
    }
    if(content==""){
        alert("내용을 작성해 주세요");
        form.content.focus();
        return false;
    }
    form.submit();
}
```

checkBoard.js



회원 관리형 게시판 프로젝트

boardList.jsp

```
<%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>
<%@ taglib uri="http://java.sun.com/jsp/jstl/functions" prefix="fn" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>게시판 목록</title>
<link rel="stylesheet" href="resources/css/style.css" />
</head>
<body>
    <c:if test="${alert=='update'}">
        <script type="text/javascript">
            alert('게시글이 수정되었습니다.');
        </c:if>
    <jsp:include page="menu.jsp" />
    <div id="container">
        <div class="title">
            <h1>게시글 목록</h1>
        </div>
        <div>
            <form class="search-form">
                <select name="field">
                    <option ${ (param.field=="title")?"selected":"" } value="title">제목</option>
                    <option ${ (param.field=="memberId")?"selected":"" } value="memberId">작성자</option>
                </select>
                <label>검색어</label>
            </form>
        </div>
    </div>
</body>
</html>
```



회원 관리형 게시판 프로젝트

boardList.jsp

```
<label>검색어</label>
<input type="text" name="text" value="${param.text}"/>
<input type="submit" value="검색" />
</form>
<table id="table2">
  <tr class="thead">
    <td>번호</td><td>글제목</td><td>글쓴이</td><td>작성일</td><td>조회수</td>
  </tr>
  <c:forEach var="board" items="${boardList}">
    <tr>
      <td>${board.bnum}</td>
      <td><a href="boardView.do?bnum=${board.bnum}">${board.title}</a></td>
      <td>${board.memberId}</td>
      <td><fmt:formatDate pattern="yyyy-MM-dd" value="${board.regDate}" /></td>
      <td><fmt:formatNumber value="${board.hit}" /> </td>
    </tr>
  </c:forEach>
  <tr>
    <td colspan="5">
      <a href="boardWriteForm.do"><input type="button" value="글쓰기"></a>
    </td>
  </tr>
</table>
```



회원 관리형 게시판 프로젝트

게시글 상세보기, 수정, 삭제

게시글 상세 보기

글 제목	진달래꽃
작성자	garlic
내용	<p>산에 진달래가 피어나기 시작하네요 봄이 왔나봐여~</p>
조회수	3
작성일자	2021-03-21 04:53:20
<div>목록수정삭제</div>	

회원 관리형 게시판 프로젝트

BoardDAO.java - 게시물 상세보기

```
//글 상세 보기
public Board getOneBoard(int num) {
    connDB();
    Board board = new Board();
    String sql = "SELECT * FROM t_board WHERE bnum = ?";
    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, num);
        rs = pstmt.executeQuery();
        rs.next();
        board.setBnum(rs.getInt("bnum"));
        board.setTitle(rs.getString("title"));
        board.setContent(rs.getString("content"));
        board.setRegDate(rs.getDate("regDate"));
        board.setMemberId(rs.getString("memberId"));
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        disconnectRS();
    }
    return board;
}
```



회원 관리형 게시판 프로젝트

BoardDAO.java - 게시물 삭제

```
//게시글 삭제하기
public boolean deleteBoard(int num) {
    connDB();
    String sql = "DELETE FROM t_board WHERE bnum=?";
    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, num);
        pstmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
        return false;
    } finally {
        disconnect();
    }
    return true;
}
```



회원 관리형 게시판 프로젝트

BoardDAO.java - 게시글 수정

```
//게시글 수정하기
public void updateBoard(Board board) {
    connDB();
    String sql = "UPDATE t_board SET title=?, content=? WHERE bnum=?";
    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, board.getTitle());
        pstmt.setString(2, board.getContent());
        pstmt.setInt(3, board.getBnum());
        pstmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {
        disconnect();
    }
}
```



회원 관리형 게시판 프로젝트

BoardDAO.java – 조회수 증가

```
//조회수 증가
public void updateHit(int bnum) {
    connDB();
    String sql = "SELECT hit FROM t_board WHERE bnum = ?";
    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, bnum);
        rs = pstmt.executeQuery();
        int hit = 0;
        if(rs.next()) { //조회수 1증가
            hit = rs.getInt("hit") + 1;
        }

        //조회수 update 쿼리
        sql = "UPDATE t_board SET hit = ? WHERE bnum = ?";
        pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, hit);
        pstmt.setInt(2, bnum);
        pstmt.executeUpdate();
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {disconnectRS();}
}
```



회원 관리형 게시판 프로젝트

boardView.jsp

```
<title>게시글 상세 보기</title>
<link rel="stylesheet" href="resources/css/style.css" />
</head>
<body>
  <jsp:include page="menu.jsp" />
  <div id="container">
    <div class="title">
      <h1>게시글 상세 보기</h1>
    </div>
    <div>
      <form action="boardUpdate.do?bnum=${board.bnum}" method="post">
        <table id="table3">
          <tr>
            <td>글 제목</td>
            <td><input type="text" name="title" value="${board.title}" class="w_title"></td>
          </tr>
          <tr>
            <td>작성자</td>
            <td>${board.memberId}</td>
          </tr>
          <tr>
            <td>내용</td>
            <td><textarea name="content">${board.content}</textarea></td>
          </tr>
        </table>
      </form>
    </div>
  </div>
</body>
```



회원 관리형 게시판 프로젝트

boardView.jsp

```
<tr>
  <td>조회수</td>
  <td><fmt:formatNumber value="${board.hit}"/></td>
</tr>
<tr>
  <td>작성일자</td>
  <td><fmt:formatDate pattern="yyyy-MM-dd hh:mm:ss" value="${board.regDate}" />
</tr>
<tr>
  <td colspan="2">
    <a href="boardList.do"><input type="button" value="목록"></a>
    <c:if test="${sessionId eq board.memberId}">
      <input type="submit" value="수정">
      <a onclick="return confirm('정말로 삭제하시겠습니까?')"
        href="boardDelete.do?bnum=${board.bnum}" class="button">
        <input type="button" value="삭제">
      </a>
    </c:if>
  </td>
</tr>
</table>
</form>
```

회원 관리형 게시판 프로젝트

MainController.java – 1 - 회원부분

```
import com.jweb2.board.Board;
import com.jweb2.board.BoardDAO;
import com.jweb2.member.Member;
import com.jweb2.member.MemberDAO;

@WebServlet("*.do") //do 확장자인 모든 파일 매핑(mapping)
public class MainController extends HttpServlet{

    private static final long serialVersionUID = 1004;

    MemberDAO memberDAO;
    BoardDAO boardDAO;

    @Override
    public void init(ServletConfig config) throws ServletException {
        memberDAO = new MemberDAO(); //memberDAO 객체 생성
        boardDAO = new BoardDAO(); //boardDAO 객체 생성
    }

    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        doPost(request, response); //get 방식인경우 처리(예. ?bnum=5)
    }
```



회원 관리형 게시판 프로젝트

MainController.java -2 - 회원부분

```
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    request.setCharacterEncoding("utf-8");
    response.setContentType("text/html; charset=utf-8");

    //클라이언트의 요청 path 정보를 추출한다.
    String uri = request.getRequestURI();
    //System.out.println(uri);
    String command = uri.substring(uri.lastIndexOf("/"));
    //System.out.println(uri.lastIndexOf("/"));

    String nextPage = null; //view인 jsp 페이지.

    if(command.equals("/memberForm.do")) { //회원 가입 요청
        nextPage="/memberForm.jsp";
    }else if(command.equals("/memberAdd.do")) { //가입 처리
        //자료 수집
        String memberId = request.getParameter("memberId");
        String passwd = request.getParameter("passwd");
        String name = request.getParameter("name");
        String gender = request.getParameter("gender");
```



회원 관리형 게시판 프로젝트

MainController.java -3 - 회원부분

```
//member 객체에 set()  
Member member = new Member();  
member.setMemberId(memberId);  
member.setPasswd(passwd);  
member.setName(name);  
member.setGender(gender);  
  
//dao - 회원 추가  
memberDAO.addMember(member);  
  
//model 저장  
request.setAttribute("member", member);  
request.setAttribute("msg", "join");  
  
//view 페이지  
nextPage="/memberResult.jsp";  
}else if(command.equals("/loginForm.do")) { //로그인 요청  
    nextPage="/loginForm.jsp";
```



회원 관리형 게시판 프로젝트

MainController.java

-4 - 회원부분

```
}else if(command.equals("/loginProcess.do")) { //로그인 처리
    //폼에 입력된 자료 수집
    String id = request.getParameter("memberId");
    String pwd = request.getParameter("passwd");

    //dao - loginCheck(), getLoginById() 호출
    int loginResult = memberDAO.loginCheck(id, pwd);
    String name = memberDAO.getLoginNameById(id);

    if(loginResult==1) {
        HttpSession session = request.getSession();
        session.setAttribute("sessionId", id);
        session.setAttribute("name", name);
        request.setAttribute("msg", "login");
    }
    //model and view
    request.setAttribute("loginResult", loginResult);
    nextPage= "/memberResult.jsp";
}else if(command.equals("/logout.do")) { //로그아웃 요청 및 세션 해제
    //session 해제
    HttpSession session = request.getSession();
    session.invalidate();
    nextPage = "/main.jsp";
```



회원 관리형 게시판 프로젝트

MainController.java -5 - 회원부분

```
}else if(command.equals("/memberView.do")) { //회원 정보 요청 및 처리
    //세션 가져오기
    HttpSession session = request.getSession();
    String sessionId = (String)session.getAttribute("sessionId");
    //dao - 회원 1명 보기
    Member member = memberDAO.getOneDB(sessionId);

    //model and view
    request.setAttribute("member", member);
    nextPage="/memberView.jsp";
}else if(command.equals("/memberDelete.do")) { //탈퇴 요청 및 처리
    String memberId = request.getParameter("memberId");
    //dao - delete
    memberDAO.deleteMember(memberId);
    request.setAttribute("msg", "delete"); //메시지 저장
    nextPage = "/memberResult.jsp";
```



회원 관리형 게시판 프로젝트

MainController.java – 6 - 회원부분

```
}else if(command.equals("/memberUpdate.do")) { //회원 수정 요청 및 처리
    //입력 폼의 자료 수집
    String memberId = request.getParameter("memberId");
    String passwd = request.getParameter("passwd");
    String name = request.getParameter("name");
    String gender = request.getParameter("gender");

    //member 객체에 set()
    Member member = new Member();
    member.setMemberId(memberId);
    member.setPasswd(passwd);
    member.setName(name);
    member.setGender(gender);

    //dao - 회원 수정
    memberDAO.updateMember(member);

    //model and view
    request.setAttribute("member", member);
    request.setAttribute("msg", "update");
    nextPage="/memberResult.jsp";
```



회원 관리형 게시판 프로젝트

MainController.java

- 7 - 게시판 부분

```
}else if(command.equals("/boardList.do")) { //게시판 목록 요청 및 처리
    //dao - getListAll()
    ArrayList<Board> boardList = boardDAO.getListAll();
    //model and view
    request.setAttribute("boardList", boardList);
    nextPage= "/boardList.jsp";
}else if(command.equals("/boardWriteForm.do")) { //글쓰기 요청
    nextPage = "/boardWriteForm.jsp";
}else if(command.equals("/boardWriteAdd.do")) { //글쓰기 처리
    //세션 및 폼 입력 자료 가져옴
    HttpSession session = request.getSession();
    String sessionId = (String)session.getAttribute("sessionId");

    String title = request.getParameter("title");
    String content = request.getParameter("content");

    //member 객체에 set()
    Board board = new Board();
    board.setTitle(title);
    board.setContent(content);
    board.setMemberId(sessionId);

    //dao - 게시글 추가
    boardDAO.insertBoard(board);

    //model and view
    request.setAttribute("board", board);
    nextPage = "/boardList.do";
```



회원 관리형 게시판 프로젝트

MainController.java – 8 – 게시판 부분

```
}else if(command.equals("/boardView.do")) { //게시글 상세 보기 및 처리
    //글번호 넘겨 받기
    int bnum = Integer.parseInt(request.getParameter("bnum"));

    //dao - 조회수
    boardDAO.updateHit(bnum);
    //dao - getOnDB() 처리,
    Board board = boardDAO.getOneBoard(bnum);

    //model and view
    request.setAttribute("board", board);
    nextPage= "/boardView.jsp";
}else if(command.equals("/boardDelete.do")) { //게시글 삭제 요청 및 처리
    //자료 수집
    int bnum = Integer.parseInt(request.getParameter("bnum"));

    //dao - delete
    boardDAO.deleteBoard(bnum);
    //model and view
    nextPage = "boardList.do";
```



회원 관리형 게시판 프로젝트

```
}else if(command.equals("/boardUpdate.do")) { //게시글 수정 요청 및 처리
    //자료 수집
    int bnum = Integer.parseInt(request.getParameter("bnum"));
    String title = request.getParameter("title");
    String content = request.getParameter("content");

    //board 객체에 set()
    Board board = new Board();
    board.setTitle(title);
    board.setContent(content);
    board.setBnum(bnum);

    //dao - 게시글 수정
    boardDAO.updateBoard(board);

    //model and view
    request.setAttribute("board", board);
    request.setAttribute("alert", "update");
    nextPage = "/boardList.do";
}

//포워딩 - view로 보내기
RequestDispatcher dispatcher = request.getRequestDispatcher(nextPage);
dispatcher.forward(request, response);
} //doPost() 닫기

} //class 닫기
```



회원 관리형 게시판 프로젝트

문자열 함수 – substring()

```
//substring()
String str = "unhappy";
String str2 = str.substring(2);
System.out.println(str2);

String word = "hamburger";
String word2 = word.substring(0, 3);
System.out.println(word2);

//indexOf()
String folder = "C:/upload";
int index = folder.indexOf("/");
System.out.println(index);
String folder2 = folder.substring(index+1);
System.out.println(folder2);

//lastIndexOf()
String url = "http://www.jweb.com/member/memberForm.jsp";
int pos = url.lastIndexOf("/");
System.out.println(pos);
String path = url.substring(pos);
System.out.println(path);
```

substring

```
public String substring(int beginIndex)
```

Returns a string that is a substring of this string. The substring begins with

Examples:

```
"unhappy".substring(2) returns "happy"
"Harbison".substring(3) returns "bison"
"emptiness".substring(9) returns "" (an empty string)
```

Parameters:

beginIndex - the beginning index, inclusive.

```
happy
ham
2
upload
26
/memberForm.jsp
```



회원 관리형 게시판 프로젝트

`getRequestUri();`

Java EE api > javax.servlet.http > HttpServletRequest

```
String getRequestedURI()
```

Returns the part of this request's URL from the protocol name up to the query string in the first line of the HTTP request. The web container does not decode this String. For example:

First line of HTTP request	Returned Value
POST /some/path.html HTTP/1.1	/some/path.html
GET http://foo.bar/a.html HTTP/1.0	/a.html
HEAD /xyz?a=b HTTP/1.1	/xyz

프로토콜 이름에서 HTTP 요청의 첫 번째 줄에 있는 쿼리 문자열까지 이 요청의 URL 부분을 반환합니다



ID 중복 검사 추가하기

회원 가입시 ID 중복 검사

회원 가입

아이디	<input type="text" value="id"/>	ID 중복
비밀번호	<input type="text" value="password"/>	
비밀번호 확인	<input type="text" value="password confirm"/>	
이름	<input type="text"/>	
성별	<input type="radio"/> 남 <input type="radio"/> 여	
<input type="button" value="등록"/> <input type="button" value="취소"/>		

localhost:8282 내용:
사용할 수 없는 ID입니다.

회원 가입

아이디	<input type="text" value="garlic"/>	ID 중복
비밀번호	<input type="text" value="password"/>	
비밀번호 확인	<input type="text" value="password confirm"/>	
이름	<input type="text"/>	
성별	<input type="radio"/> 남 <input type="radio"/> 여	
<input type="button" value="등록"/> <input type="button" value="취소"/>		

- ▼ Jweb2
 - > Deployment Descriptor: Jweb2
 - > JAX-WS Web Services
 - ▼ Java Resources
 - ▼ src
 - ▼ com.jweb2.board
 - > Board.java
 - > BoardDAO.java
 - ▼ com.jweb2.controller
 - > MainController.java
 - ▼ com.jweb2.member
 - > IDcheckServlet.java
 - > Member.java
 - > MemberDAO.java
 - > Libraries
 - > build
 - ▼ WebContent
 - > META-INF
 - > resources
 - > WEB-INF
 - boardList.jsp
 - boardView.jsp

ID 중복 검사

✓ id 중복 검사 메서드 만들기

MemberDAO.java

```
//ID 중복검사
public boolean duplicatedID(String memberId) {
    connDB();
    boolean result = false;
    String sql = "SELECT DECODE(count(*), 1, 'true', 'false') AS result "
        + "FROM t_member WHERE memberId = ?";

    try {
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, memberId);
        rs = pstmt.executeQuery();
        if(rs.next()) {
            result = rs.getBoolean("result");
        }
    } catch (SQLException e) {
        e.printStackTrace();
    } finally {disconnectRS();}
    return result;
}
```



ID 중복 검사

✓ id 중복 버튼 추가하기

memberForm.jsp

```
<tr>
  <td>아이디</td>
  <td class="t_id">
    <input type="text" id="t_id" name="memberId" placeholder="id">
    <input type="button" value="ID 중복" onClick="fn_process()">
  </td>
</tr>
```

ID 중복 검사

✓ Id 중복 검사 ajax 만들기

```
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script type="text/javascript">
    function fn_process(){
        var t_id = $("#t_id").val();
        $.ajax({
            type:"post",
            dataType:"text",
            async:false,
            url:"http://localhost:8282/Jweb2/idcheck",
            data: {id:t_id}, //id를 서블릿으로 전송
            success:function(data, textStatus){
                if(data=='usable'){
                    alert("사용할 수 있는 ID입니다.")
                }else{
                    alert("사용할 수 없는 ID입니다.")
                }
            },
            error:function(data, textStatus){
                alert('에러 발생');
            },
        });
    }
</script>
```

memberForm.jsp



ID 중복 검사

✓ Idcheck 서블릿(매핑 주소)

```
@WebServlet("/idcheck")
public class IDcheckServlet extends HttpServlet {
    @Override
    protected void doPost(HttpServletRequest request, HttpServletResponse response) {
        request.setCharacterEncoding("utf-8");
        response.setContentType("text/html; charset=utf-8");
        PrintWriter out = response.getWriter();
        //ajax의 id 넘겨 받기
        String id = request.getParameter("id");

        //dao - duplicatedID() 호출
        MemberDAO memberDAO = new MemberDAO();
        boolean duplicatedID = memberDAO.duplicatedID(id);

        if(duplicatedID == true) { //결과 메시지를 회원가입 폼(ajax)에 전송
            out.print("not_usable");
        } else {
            out.print("usable");
        }
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response) {
        doPost(request, response);
    }
}
```

IDcheckServlet.java

