

7장. 회원 및 게시판 관리

Member & Board



회원 관리 프로젝트

- 회원관리 프로젝트
 - 프로젝트 이름 : Members

```
▼ <img alt="Java icon" data-bbox="205 305 225 325"/> > Members [javaweb main]
  > <img alt="Deployment Descriptor icon" data-bbox="225 340 245 360"/> Deployment Descriptor: Members
  > <img alt="JAX-WS icon" data-bbox="225 375 245 395"/> JAX-WS Web Services
  ▼ <img alt="Java Resources icon" data-bbox="225 410 245 430"/> Java Resources
    ▼ <img alt="src/main/java icon" data-bbox="245 445 265 465"/> > src/main/java
      > <img alt="board folder icon" data-bbox="265 480 285 500"/> > board
      ▼ <img alt="common folder icon" data-bbox="265 515 285 535"/> > common
        > <img alt="DBConnectionTest.java icon" data-bbox="285 550 305 570"/> > DBConnectionTest.java
        > <img alt="DBUtil.java icon" data-bbox="285 585 305 605"/> > DBUtil.java
      ▼ <img alt="member folder icon" data-bbox="265 620 285 640"/> > member
        ▼ <img alt="controller folder icon" data-bbox="285 655 305 675"/> > controller
          > <img alt="MemberController.java icon" data-bbox="305 690 325 710"/> > MemberController.java
        ▼ <img alt="model folder icon" data-bbox="285 725 305 745"/> > model
          > <img alt="Member.java icon" data-bbox="305 760 325 780"/> > Member.java
          > <img alt="MemberDAO.java icon" data-bbox="305 795 325 815"/> > MemberDAO.java
        ▼ <img alt="service folder icon" data-bbox="285 830 305 850"/> > service
          > <img alt="MemberService.java icon" data-bbox="305 865 325 885"/> > MemberService.java
```

java

```
▼ <img alt="webapp folder icon" data-bbox="575 315 595 335"/> > webapp
  ▼ <img alt="board folder icon" data-bbox="595 350 615 370"/> > board
    > <img alt="boardList.jsp icon" data-bbox="615 385 635 405"/> > boardList.jsp
    <img alt="detail.jsp icon" data-bbox="615 420 635 440"/> detail.jsp
    <img alt="updateForm.jsp icon" data-bbox="615 455 635 475"/> updateForm.jsp
    <img alt="write.jsp icon" data-bbox="615 490 635 510"/> write.jsp
  > <img alt="member folder icon" data-bbox="595 535 615 555"/> > member
  > <img alt="META-INF folder icon" data-bbox="595 570 615 590"/> META-INF
  ▼ <img alt="resources folder icon" data-bbox="595 605 615 625"/> > resources
    ▼ <img alt="css folder icon" data-bbox="615 640 635 660"/> > css
      <img alt="style.css icon" data-bbox="635 675 655 695"/> > style.css
    > <img alt="images folder icon" data-bbox="615 710 635 730"/> > images
  > <img alt="WEB-INF folder icon" data-bbox="595 745 615 765"/> WEB-INF
    <img alt="index.jsp icon" data-bbox="605 780 625 800"/> index.jsp
```

web



Maven 기반 프로젝트

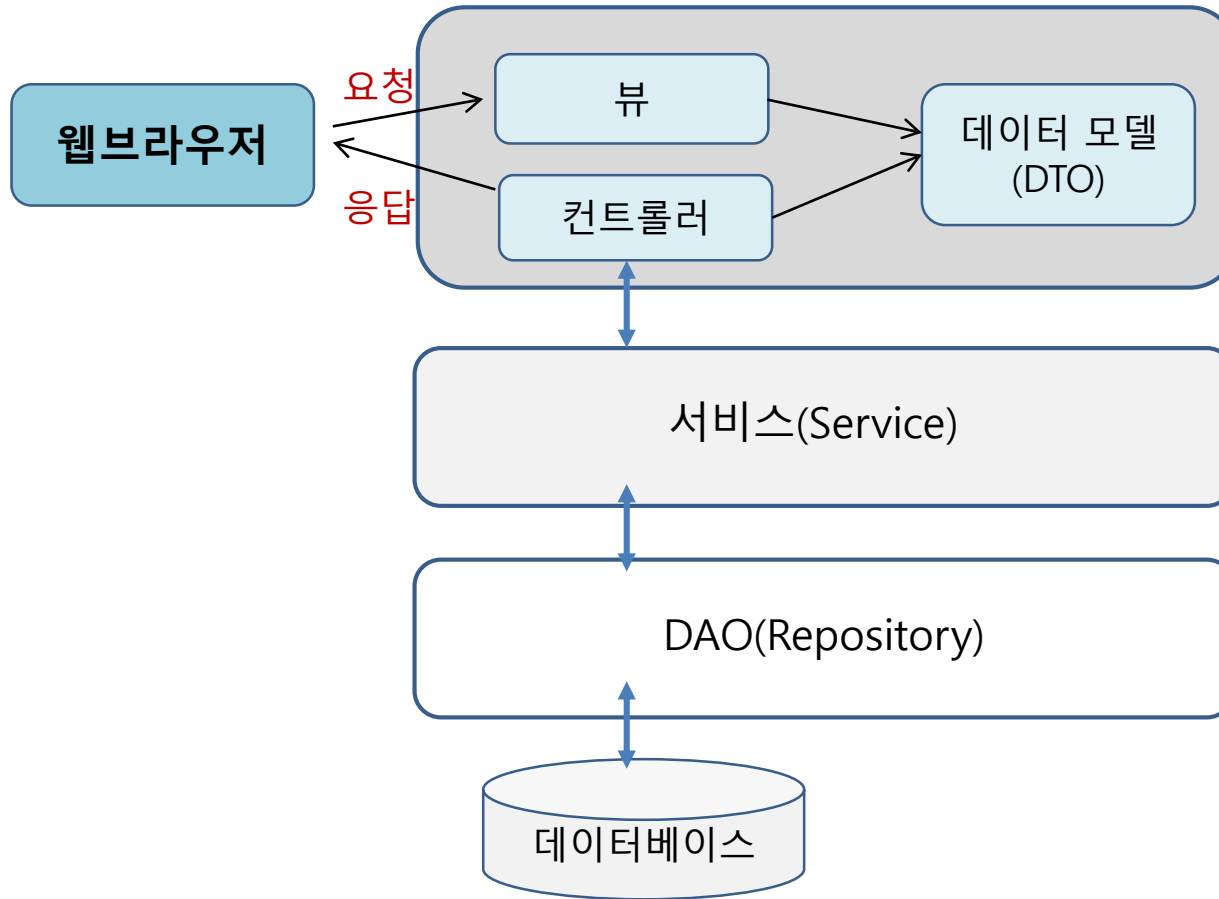
1. 프로젝트 > 우클릭 > configure > Convert to Maven Project
2. pom.xml 에 <dependencies> 의존성 관리 (3개 추가)

```
<dependencies>
  <dependency>
    <groupId>jakarta.servlet.jsp.jstl</groupId>
    <artifactId>jakarta.servlet.jsp.jstl-api</artifactId>
    <version>3.0.0</version>
  </dependency>
  <dependency>
    <groupId>org.glassfish.web</groupId>
    <artifactId>jakarta.servlet.jsp.jstl</artifactId>
    <version>3.0.1</version>
  </dependency>
  <dependency>
    <groupId>com.mysql</groupId>
    <artifactId>mysql-connector-j</artifactId>
    <version>8.4.0</version>
  </dependency>
</dependencies>
```



회원 관리 프로젝트

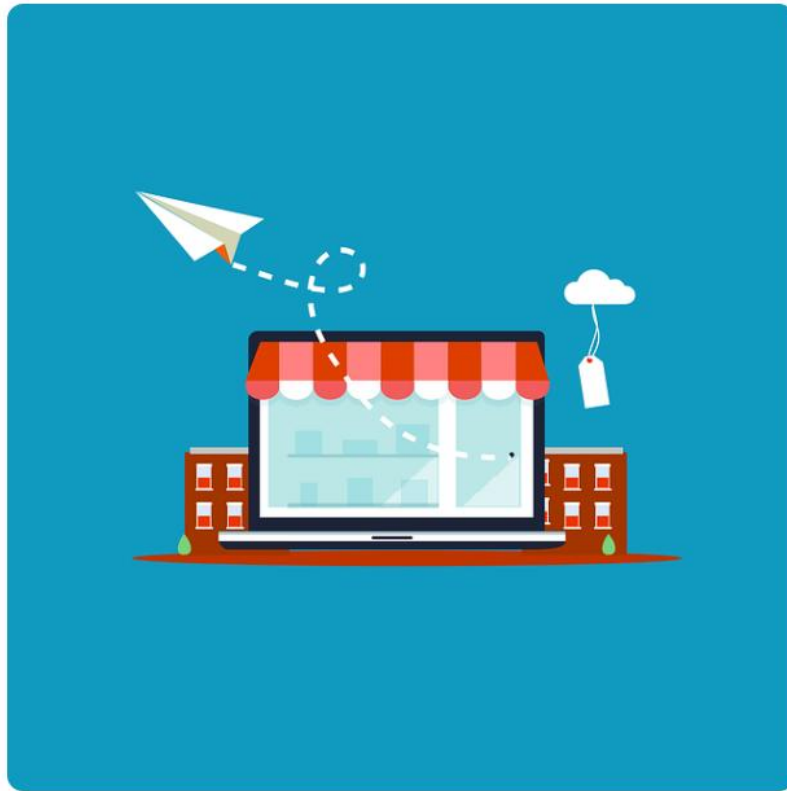
■ 계층적 구조



회원 관리 프로젝트

- 인덱스 페이지

Good Mall에 오신 것을 환영합니다.



[회원 가입](#) | [로그인](#)

회원 관리 프로젝트

■ 인덱스 페이지

```
<head>
  <meta charset="UTF-8">
  <title>Welcome~ Good Mall...</title>
  <link rel="stylesheet" href="/resources/css/style.css">
</head>
<body>
  <section id="container">
    <div id="main">
      <h2>Good Mall에 오신 것을 환영합니다.</h2>

      <div class="main-pic">
        
      </div>

      <p>
        <a href="/member?action=addForm">회원 가입</a> &nbsp; | &nbsp;
        <a href="/member?action=loginForm">로그인</a> &nbsp;
      </p>
    </div>
  </section>
</body>
```



회원 관리 프로젝트

■ style.css

```
#container{
    width: 80%;
    margin: 20px auto;
}
#main{text-align: center;}
a{text-decoration: none;}
a:hover{text-decoration: underline; }
ul li{margin: 10px;}
h2{
    border-bottom: 2px solid #ccc;
    padding-bottom: 10px;
}
.main-pic img{
    width: 40%;
    border-radius: 10px;
}
table{
    border: 1px solid #ccc;
    border-collapse: collapse;
    margin-bottom: 10px;
}
```



회원 관리 프로젝트

■ style.css

```
table th, td{
    border: 1px solid #ccc;
    padding: 5px 10px;
}
ul li{list-style: none;}
ul li label{
    width: 80px;
    float: left;
}
/* login 스타일 */
.btnLogin, .btnJoin{margin-left: 20px;}

/* write 스타일 */
.writeForm, .detail{
    width: 500px;
}
.writeForm ul, .detail ul{padding: 0;}
textarea{resize: none;}
.btnWrite{
    margin: 10px 10px 10px 350px;
}
.btnWrite button{padding: 5px;}
```



회원 관리 프로젝트

▪ MemberDTO

```
public class Member {  
    private String mid;  
    private String passwd;  
    private String name;  
    private String gender;  
    private Timestamp joinDate;  
  
    public Member() {} //기본 생성자  
  
    public Member(String mid, String passwd, String name,  
        String gender, Timestamp joinDate) {  
        this.mid = mid;  
        this.passwd = passwd;  
        this.name = name;  
        this.gender = gender;  
        this.joinDate = joinDate;  
    }  
}
```



회원 관리 프로젝트

▪ MemberDTO

```
public String getMid() {  
    return mid;  
}  
public void setMid(String mid) {  
    this.mid = mid;  
}  
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;  
}
```



회원 관리 프로젝트

▪ MemberDTO

```
public String getGender() {  
    return gender;  
}  
public void setGender(String gender) {  
    this.gender = gender;  
}  
public Timestamp getJoinDate() {  
    return joinDate;  
}  
public void setJoinDate(Timestamp joinDate) {  
    this.joinDate = joinDate;  
}  
}
```

회원 관리 프로젝트

■ jspdb > Member 테이블

```
-- member 테이블 생성
create table member(
    mid      varchar(20) primary key,
    passwd   varchar(20) not null,
    name     varchar(30) not null,
    gender   varchar(10),
    joindate timestamp default now()
);

insert into member values ('today12', 'm1234', '김기용', '남자', now());
```

mid	passwd	name	gender	joindate
today12	m1234	김기용	남자	2025-10-13 17:04:18

회원 관리 프로젝트

▪ JDBC 연결 테스트

```
public class DBConnectionTest {  
    private static final String URL = "jdbc:mysql://localhost:3306/jspdb";  
    private static final String USER = "javauser";  
    private static final String PASSWORD = "pwjava";  
  
    public static void main(String[] args) {  
  
        try{  
            Connection conn = DriverManager.getConnection(URL, USER, PASSWORD);  
            System.out.println(conn + "DB 접속 성공!!");  
        }catch(Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

```
com.mysql.cj.jdbc.ConnectionImpl@58ea606cDB 접속 성공!!
```



회원 관리 프로젝트

▪ DBUtil 클래스

```
public class DBUtil {  
    private static final String URL = "jdbc:mysql://localhost:3306/jspdb";  
    private static final String USER = "javauser";  
    private static final String PASSWORD = "pwjava";  
  
    public static Connection getConnection() {  
  
        try{  
            Class.forName("com.mysql.cj.jdbc.Driver");  
            return DriverManager.getConnection(URL, USER, PASSWORD);  
        }catch(Exception e) {  
            e.printStackTrace();  
            return null;  
        }  
    }  
}
```

회원 관리 프로젝트

▪ MemberDAO – 회원 가입

```
public void addMember(Member member) {  
    String sql = "insert into member(mid, passwd, name, gender, joindate) "  
        + "values (?, ?, ?, ?, now())";  
  
    try(Connection conn = DBUtil.getConnection();  
        PreparedStatement pstmt = conn.prepareStatement(sql)){  
        pstmt.setString(1, member.getMid());  
        pstmt.setString(2, member.getPasswd());  
        pstmt.setString(3, member.getName());  
        pstmt.setString(4, member.getGender());  
  
        pstmt.executeUpdate(); //실행  
    }catch(Exception e) {  
        e.printStackTrace();  
    }  
}
```

회원 관리 프로젝트

▪ MemberDAO – 회원 목록

```
public List<Member> getMemberList(){
    List<Member> list = new ArrayList<>();
    String sql = "select * from member";

    try(Connection conn = DBUtil.getConnection();
        PreparedStatement pstmt = conn.prepareStatement(sql);
        ResultSet rs = pstmt.executeQuery()){

        while(rs.next()) {
            Member member = new Member(
                rs.getString("mid"),
                rs.getString("passwd"),
                rs.getString("name"),
                rs.getString("gender"),
                rs.getTimestamp("joindate")
            );
            list.add(member);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return list;
}
```



회원 관리 프로젝트

▪ MemberDAO – 회원 상세(정보)

```
public Member getMember(String mid) {
    Member member = new Member();
    String sql = "select * from member where mid = ?";

    try(Connection conn = DBUtil.getConnection();
        PreparedStatement pstmt = conn.prepareStatement(sql)){
        pstmt.setString(1, mid);

        ResultSet rs = pstmt.executeQuery();
        if(rs.next()) {
            member.setMid(rs.getString("mid"));
            member.setPasswd(rs.getString("passwd"));
            member.setName(rs.getString("name"));
            member.setGender(rs.getString("gender"));
            member.setJoinDate(rs.getTimestamp("joindate"));
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return member;
}
```



회원 관리 프로젝트

▪ MemberDAO – 로그인

```
public boolean checkLogin(Member member) {  
  
    String sql = "select * from member where mid=? and passwd=?";  
    try(Connection conn = DBUtil.getConnection();  
        PreparedStatement pstmt = conn.prepareStatement(sql)){  
        pstmt.setString(1, member.getMid());  
        pstmt.setString(2, member.getPasswd());  
  
        ResultSet rs = pstmt.executeQuery();  
        if(rs.next()) {  
            return true;  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return false;  
}
```



회원 관리 프로젝트

- 회원 가입

회원 가입

아이디	<input type="text" value="coli"/>
비밀번호	<input type="password" value="....."/>
이름	<input type="text" value="우영우"/>
성별	<input type="radio"/> 남 <input checked="" type="radio"/> 여

회원 관리 프로젝트

▪ addMember.jsp

```
<section id="container">
  <h2>회원 가입</h2>
  <form action="/member?action=add" method="post">
    <ul>
      <li>
        <label>아이디</label>
        <input type="text" name="mid">
      </li>
      <li>
        <label>비밀번호</label>
        <input type="password" name="passwd">
      </li>
      <li>
        <label>이름</label>
        <input type="text" name="name">
      </li>
    </ul>
  </form>
</section>
```

회원 관리 프로젝트

▪ addMember.jsp

```
<li>
    <label>성별</label>
    <input type="radio" name="gender" value="남자" checked>남
    <input type="radio" name="gender" value="여자">여
</li>
</ul>
<div class="btn">
    <input type="submit" value="가입">
    <input type="reset" value="취소">
</div>
</form>
```

회원 관리 프로젝트

- 회원 목록

회원 목록

[Home](#) | [\[today12\]님 로그아웃](#)

번호	아이디	이름	성별	가입일
1	coli	우영우	여자	2025-10-15 16:54:43.0
2	today12	김기용	남자	2025-10-13 17:04:18.0

회원 관리 프로젝트

▪ memberList.jsp

```
<%@ taglib uri="jakarta.tags.core" prefix="c" %>
```

```
<section id="container">
  <h2>회원 목록</h2>
  <p>
    <a href="/index.jsp">Home</a> &nbsp; | &nbsp;
    [${sessionId}]님 <a href="/member?action=logout">로그아웃</a>
  </p>
  <table>
    <thead>
      <tr>
        <th>번호</th>
        <th>아이디</th>
        <th>이름</th>
        <th>성별</th>
        <th>가입일</th>
      </tr>
    </thead>
```



회원 관리 프로젝트

▪ memberList.jsp

```
<tbody>
  <c:forEach var="member" items="#{memberList}" varStatus="i">
    <tr>
      <td>${i.count}</td>
      <td>
        <a href="/member?action=info&mid=${member.mid}">${member.mid}</a>
      </td>
      <td>${member.name}</td>
      <td>${member.gender}</td>
      <td>${member.joinDate}</td>
    </tr>
  </c:forEach>
</tbody>
</table>
```


회원 관리 프로젝트

■ 회원 상세 보기

회원 정보

- 아이디: today12
- 비밀번호: m1234
- 이름: 김기용
- 성별: 남자
- 가입일: 2025-10-13 17:04:18.0

목록

회원 관리 프로젝트

▪ memberInfo.jsp

```
<link rel="stylesheet" href="../../resources/css/style.css">
</head>
<body>
    <section id="container">
        <h2>회원 정보</h2>
        <ul>
            <li>아이디: ${member.mid}</li>
            <li>비밀번호: ${member.passwd}</li>
            <li>이름: ${member.name}</li>
            <li>성별: ${member.gender}</li>
            <li>가입일: ${member.joinDate}</li>
        </ul>
        <a href="/member?action=list">
            <button type="button">목록</button>
        </a>
    </section>
</body>
```



회원 관리 프로젝트

- 로그인

로그인

아이디

coli

비밀번호

.....

로그인

취소

회원 관리 프로젝트

▪ login.jsp

```
<h2>로그인</h2>
<form action="/member?action=login" method="post">
  <ul>
    <li>
      <label>아이디</label>
      <input type="text" name="mid">
    </li>
    <li>
      <label>비밀번호</label>
      <input type="password" name="passwd">
    </li>
  </ul>
  <div style="color: red">
    <c:if test="${not empty error}">
      ${error}
    </c:if>
  </div>
  <div class="btn">
    <input type="submit" value="로그인">
    <input type="reset" value="취소">
  </div>
</form>
```



회원 관리 프로젝트

▪ MemberService – 서비스 계층

```
public class MemberService {  
  
    MemberDAO dao = new MemberDAO();  
  
    //회원 가입  
    public void addMember(Member member) {  
        dao.addMember(member);  
    }  
  
    //회원 목록  
    public List<Member> getMemberList(){  
        return dao.getMemberList();  
    }  
  
    //로그인  
    public boolean checkLogin(Member member) {  
        return dao.checkLogin(member);  
    }  
  
    //회원 정보  
    public Member getMember(String mid) {  
        return dao.getMember(mid);  
    }  
}
```



회원 관리 프로젝트

▪ MemberController – 서블릿

```
@WebServlet("/member")
public class MemberController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    MemberService service; //인스턴스 선언

    public MemberController() { //인스턴스 생성
        service = new MemberService();
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse
        String action = request.getParameter("action");
        String nextPage = "";
        HttpSession session = request.getSession();

        if(action.equals("list")) {
            List<Member> memberList = service.getMemberList();
            request.setAttribute("memberList", memberList);
            nextPage = "/member/memberList.jsp";
        } else if(action.equals("addForm")) {
            nextPage = "/member/addMember.jsp";
        }
    }
}
```



회원 관리 프로젝트

▪ MemberController – 서블릿

```
}else if(action.equals("add")) {  
    String mid = request.getParameter("mid");  
    String pwd = request.getParameter("passwd");  
    String name = request.getParameter("name");  
    String gender = request.getParameter("gender");  
  
    Member member = new Member();  
    member.setMid(mid);  
    member.setPasswd(pwd);  
    member.setName(name);  
    member.setGender(gender);  
  
    service.addMember(member); //addMember() 호출  
    session.setAttribute("sessionId", mid); //세션 발급  
  
    //등록 후 목록 페이지로 이동  
    response.sendRedirect("/member?action=list");  
    return; //즉시 종료  
}else if(action.equals("loginForm")) {  
    nextPage = "/member/login.jsp";
```



회원 관리 프로젝트

▪ MemberController – 서블릿

```
}else if(action.equals("login")) {  
    String id = request.getParameter("mid");  
    String pw = request.getParameter("passwd");  
  
    Member member = new Member();  
    member.setMid(id);  
    member.setPasswd(pw);  
  
    boolean result = service.checkLogin(member);  
    if(result) {  
        session.setAttribute("sessionId", id); //세션 발급  
        response.sendRedirect("/member?action=list");  
        return;  
    }else {  
        String error = "아이디나 비밀번호가 일치하지 않습니다.";  
        request.setAttribute("error", error);  
        nextPage = "/member/login.jsp";  
    }  
}
```



회원 관리 프로젝트

▪ MemberController – 서블릿

```
}else if(action.equals("logout")) {  
    session.invalidate();  
    nextPage = "index.jsp";  
}else if(action.equals("info")) {  
    String mid = request.getParameter("mid");  
  
    Member member = service.getMember(mid);  
    request.setAttribute("member", member);  
    nextPage = "/member/memberInfo.jsp";  
}  
  
// 포워딩  
RequestDispatcher rd =  
    request.getRequestDispatcher(nextPage);  
rd.forward(request, response);  
}
```



게시판 관리 프로젝트

- 인덱스 페이지

Good Mall에 오신 것을 환영합니다.



[회원 가입](#) | [로그인](#) | [게시판](#)

게시판 관리 프로젝트

▪ BoardDTO

```
public class Board {  
    private int bnum;        //글번호  
    private String title;    //글제목  
    private String content;  //글내용  
    private String mid;      //글쓴이(Member join)  
    private Timestamp regDate; //작성일  
  
    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;  
    }  
}
```



게시판 관리 프로젝트

▪ BoardDTO

```
public String getContent() {  
    return content;  
}  
public void setContent(String content) {  
    this.content = content;  
}  
public String getMid() {  
    return mid;  
}  
public void setMid(String mid) {  
    this.mid = mid;  
}  
public Timestamp getRegDate() {  
    return regDate;  
}  
public void setRegDate(Timestamp regDate) {  
    this.regDate = regDate;  
}  
}
```



게시판 관리 프로젝트

- MySQL - 테이블

```
-- board 테이블 생성
create table board(
    bnum    int primary key auto_increment, -- 자동 순번
    title   varchar(100) not null,
    content text,
    mid     varchar(20),
    regdate timestamp default now(),
    foreign key (mid) references member(mid) -- 외래키
    on delete cascade
);

-- 게시글 삽입
insert into board (title, content, mid, regdate)
values ('가입 인사', '안녕하세요~ 가입 해요..', 'today12', now());
```

게시판 관리 프로젝트

▪ BoardDAO – 글 목록 보기

```
public List<Board> getBoardList(){
    List<Board> boardList = new ArrayList<>();
    String sql = "select * from board order by bnum desc";

    try(Connection conn = DBUtil.getConnection();
        PreparedStatement ps = conn.prepareStatement(sql);
        ResultSet rs = ps.executeQuery()){

        while(rs.next()) {
            Board board = new Board();
            board.setBnum(rs.getInt("bnum"));
            board.setTitle(rs.getString("title"));
            board.setContent(rs.getString("content"));
            board.setMid(rs.getString("mid"));
            board.setRegDate(rs.getTimestamp("regdate"));
            boardList.add(board);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
    return boardList;
}
```



게시판 관리 프로젝트

▪ BoardDAO – 글 등록

```
public void addBoard(Board board) {
    String sql = "insert into board(title, content, mid, regdate) "
        + "values (?, ?, ?, now())";

    try(Connection conn = DBUtil.getConnection();
        PreparedStatement ps = conn.prepareStatement(sql)){
        ps.setString(1, board.getTitle());
        ps.setString(2, board.getContent());
        ps.setString(3, board.getMid());
        ps.executeUpdate();
    }catch(Exception e) {
        e.printStackTrace();
    }
}
```

게시판 관리 프로젝트

▪ BoardDAO – 글 상세 보기

```
public Board getBoard(int bnum) {  
    Board board = new Board();  
    String sql = "select * from board where bnum = ?";  
  
    try(Connection conn = DBUtil.getConnection();  
        PreparedStatement ps = conn.prepareStatement(sql)){  
        ps.setInt(1, bnum);  
  
        ResultSet rs = ps.executeQuery();  
        if(rs.next()) {  
            board.setBnum(rs.getInt("bnum"));  
            board.setTitle(rs.getString("title"));  
            board.setContent(rs.getString("content"));  
            board.setMid(rs.getString("mid"));  
            board.setRegDate(rs.getTimestamp("regdate"));  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return board;  
}
```



게시판 관리 프로젝트

▪ BoardService – 서비스 계층

```
public class MemberService {  
    MemberDAO dao = new MemberDAO();  
  
    //회원 가입  
    public void addMember(Member member) {  
        dao.addMember(member);  
    }  
  
    //회원 목록  
    public List<Member> getMemberList(){  
        return dao.getMemberList();  
    }  
  
    //로그인  
    public boolean checkLogin(Member member) {  
        return dao.checkLogin(member);  
    }  
  
    //회원 정보  
    public Member getMember(String mid) {  
        return dao.getMember(mid);  
    }  
}
```



게시판 관리 프로젝트

▪ BoardController – 컨트롤러 계층

```
@WebServlet("/board")
public class BoardController extends HttpServlet {
    private static final long serialVersionUID = 1L;

    BoardService service = new BoardService();

    protected void doGet(HttpServletRequest request, HttpServletResponse
        doPost(request, response);
    }

    protected void doPost(HttpServletRequest request, HttpServletResponse
        String action = request.getParameter("action");
        String nextPage = "";
        HttpSession session = request.getSession(); //세션 생성
```

게시판 관리 프로젝트

▪ BoardController – 컨트롤러 계층

```
if("boardList".equals(action)) { //목록 보기
    List<Board> boardList = service.getBoardList();
    request.setAttribute("boardList", boardList);
    nextPage = "/board/boardList.jsp";
}else if("writeForm".equals(action)) { //글쓰기 페이지
    nextPage = "/board/write.jsp";
}else if("write".equals(action)) { //글쓰기 처리
    String title = request.getParameter("title");
    String content = request.getParameter("content");
    String mid = (String)session.getAttribute("sessionId");

    Board board = new Board();
    board.setTitle(title);
    board.setContent(content);
    board.setMid(mid);

    service.addBoard(board); //등록 메서드 호출
    //글쓰기 후 글 목록으로 이동
    response.sendRedirect("/board?action=boardList");
    return;
}
```



게시판 관리 프로젝트

▪ BoardController – 컨트롤러 계층

```
}else if("detail".equals(action)) { //글 상세보기
    int bnum = Integer.parseInt(request.getParameter("bnum"));

    Board board = service.getBoard(bnum);

    request.setAttribute("board", board);
    nextPage = "/board/detail.jsp";
}

//포워딩
RequestDispatcher rd =
    request.getRequestDispatcher(nextPage);
rd.forward(request, response);
}
```

게시판 관리 프로젝트

- 게시글 목록

게시글 목록

[홈으로](#)

글번호	글제목	글쓴이	작성일
3	K-move	today12	2025-10-16 18:25:40.0
2	가을 풍경	coli	2025-10-16 17:12:28.0
1	가입 인사	today12	2025-10-16 17:10:37.0

글쓰기

게시판 관리 프로젝트

■ 게시물 목록

```
<head>
  <meta charset="UTF-8">
  <title>게시판 목록</title>
  <link rel="stylesheet" href="../../resources/css/style.css">
</head>
<body>
  <section id="container">
    <h2>게시글 목록</h2>
    <p><a href="/index.jsp">홈으로</a><p>
    <table>
      <thead>
        <tr>
          <th>글번호</th>
          <th>글제목</th>
          <th>글쓴이</th>
          <th>작성일</th>
        </tr>
      </thead>
```



게시판 관리 프로젝트

■ 게시물 목록

```
<tbody>
  <c:forEach var="board" items="${boardList}">
    <tr>
      <td>${board.bnum}</td>
      <td><a href="/board?action=detail&bnum=${board.bnum}">
        ${board.title}</a>
      </td>
      <td>${board.mid}</td>
      <td><fmt:formatDate value="${board.regDate}"
        pattern="yyyy-MM-dd HH:mm:ss" /></td>
    </tr>
  </c:forEach>
</tbody>
</table>
<div class="btnWrite">
  <a href="/board?action=writeForm" >
    <button type="button">글쓰기</button>
  </a>
</div>
</section>
```



게시판 관리 프로젝트

■ 게시글 등록

글쓰기는 로그인한 회원만 가능함

localhost:8080 내용:
로그인이 필요합니다.

확인

로그인

아이디

비밀번호

게시판 관리 프로젝트

- 게시글 등록

글쓰기

해외 취업
풀 스택 웹 개발자 과정

등록

게시판 관리 프로젝트

■ 게시물 등록

```
<section id="container">
  <c:if test="${empty sessionId }">
    <script>
      alert("로그인이 필요합니다.");
      location.href="/member?action=loginForm";
    </script>
  </c:if>

  <h2>글쓰기</h2>
  <form action="/board" method="post" class="writeForm">
    <input type="hidden" name="action" value="write">
    <fieldset>
      <ul>
        <li>
          <input type="text" name="title" size=62
            placeholder="제목">
        </li>
      </ul>
    </fieldset>
  </form>
</section>
```



게시판 관리 프로젝트

■ 게시물 등록

```
<li>
    <textarea name="content" rows="10" cols="60"
        placeholder="내용"></textarea>
</li>
<li>
    <input type="text" name="mid" size=62
        value="${sessionId}">
</li>
</ul>
</fieldset>
<p><button type="submit">등록</button></p>
</form>
</section>
```

게시판 관리 프로젝트

- 게시글 상세 보기

게시글 상세보기

가을 옷을 준비

벌써 겨울이 오고있네

글쓴이: today12 (작성일: 2025-10-20 06:03:34.0)

수정

삭제

목록

게시판 관리 프로젝트

■ 게시물 상세 보기

```
<h2>게시글 상세보기</h2>
<div class="detail">
  <fieldset>
    <ul>
      <li>
        <input type="text" name="title" size=62
          value="${board.title}" readonly>
      </li>
      <li>
        <textarea name="content" rows="10" cols="60"
          readonly>${board.content}</textarea>
      </li>
      <li>
        <input type="text" name="mid" size=62
          value="글쓴이: ${board.mid} (작성일: ${board.regDate})">
      </li>
    </ul>
  </fieldset>
</div>
```



게시판 관리 프로젝트

- 게시글 상세 보기

```
<p>
  <c:if test="${board.mid eq sessionId}">
    <a href="/board?action=updateForm&bnum=${board.bnum}">
      <button>수정</button></a>
    <a href="/board?action=delete&bnum=${board.bnum}">
      <button onclick="return confirm('정말로 삭제하시겠습니까?')">삭제</button></a>
    </c:if>
    <a href="/board?action=boardList"><button>목록</button></a>
  </p>
```