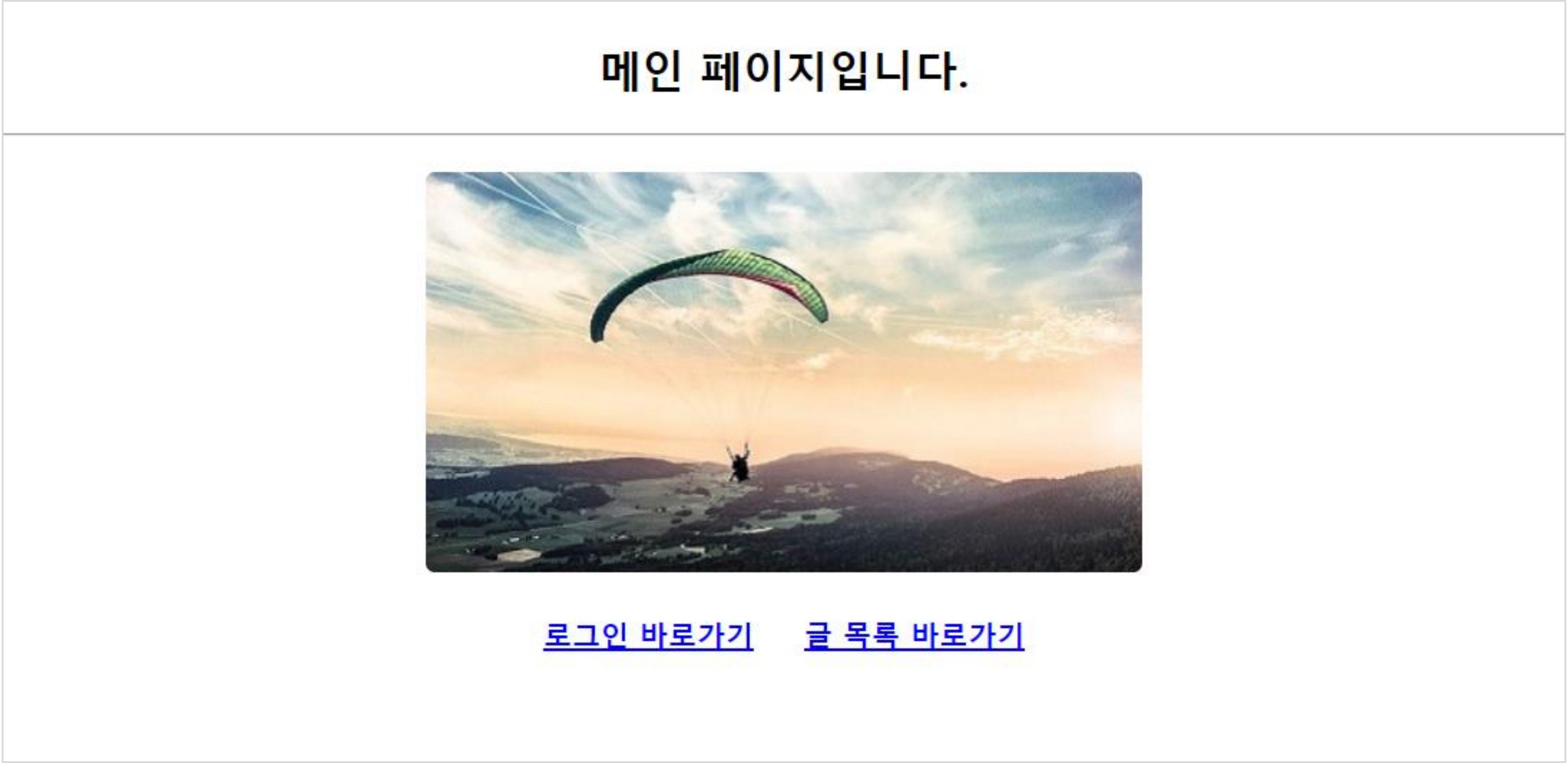
8장. SpringBoard2 프로젝트

SpringBoard2

SpringBoard

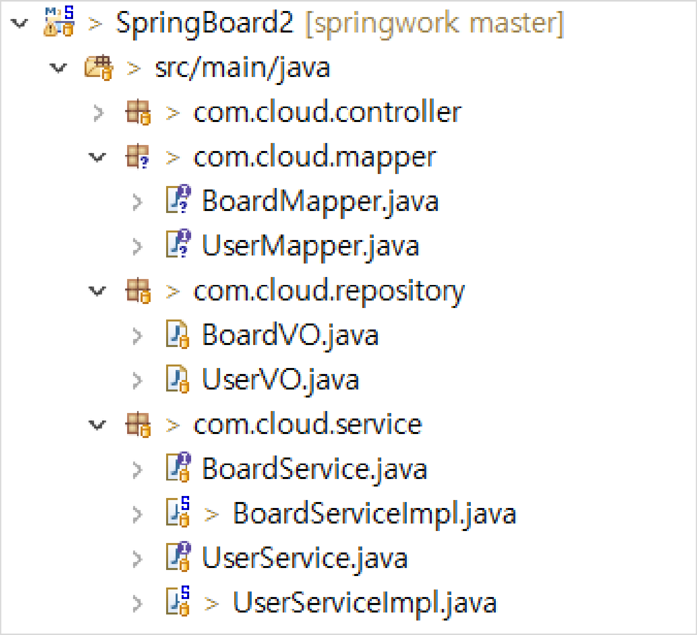


메인 페이지입니다.

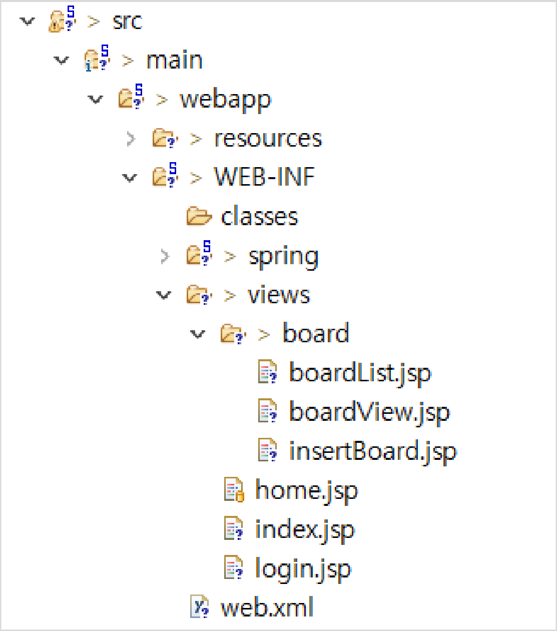
로그인 바로가기 글 목록 바로가기

SpringBoard

- 프로젝트 구조도

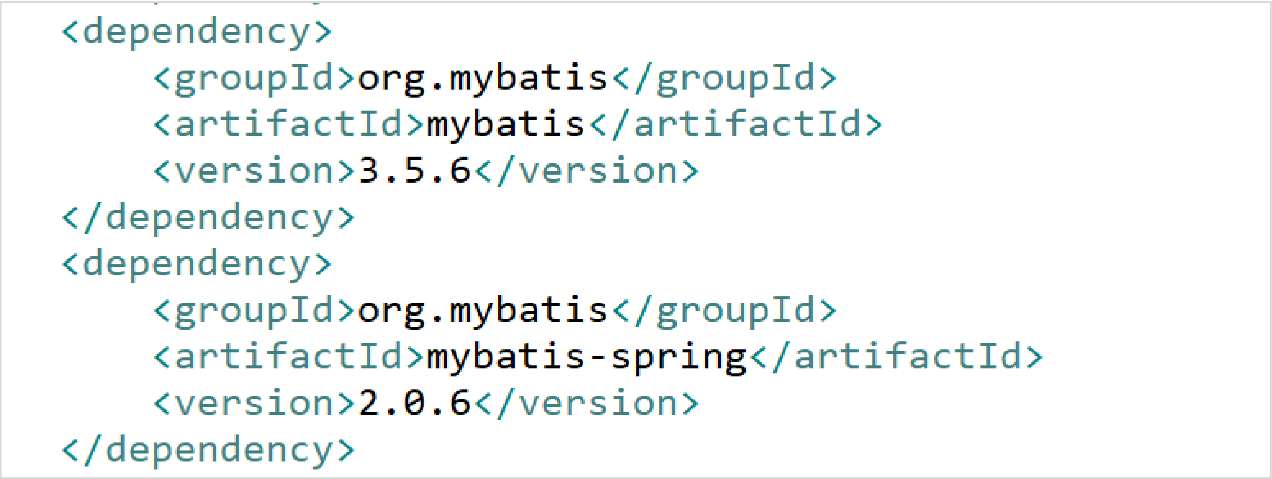


```
v > SpringBoard2 [springwork master]
    v > src/main/java
        > > com.cloud.controller
        v > com.cloud.mapper
            > BoardMapper.java
            > UserMapper.java
        v > com.cloud.repository
            > BoardVO.java
            > UserVO.java
        v > com.cloud.service
            > BoardService.java
            > > BoardServiceImpl.java
            > UserService.java
            > > UserServiceImpl.java
```

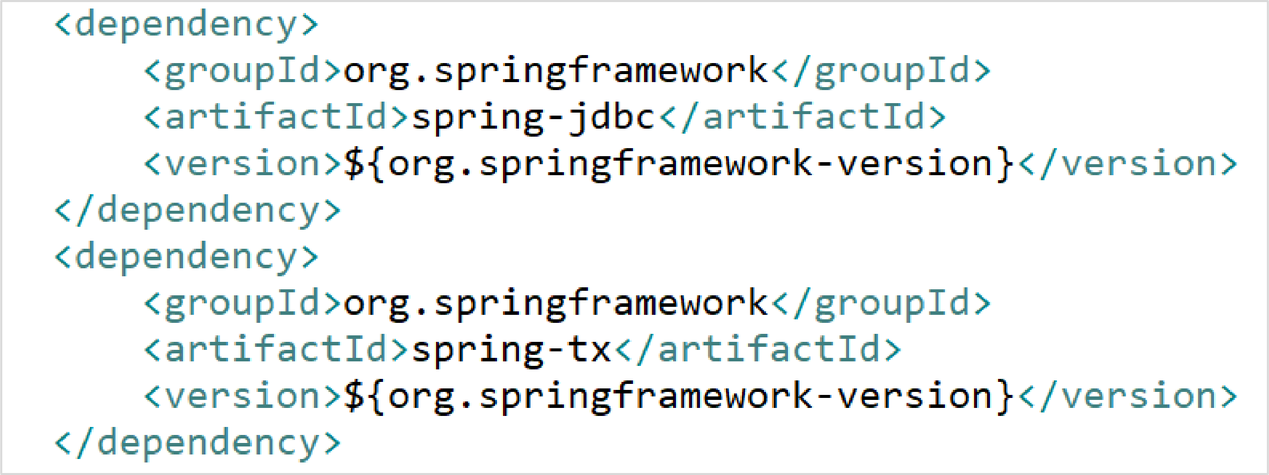


```
v > src
    v > main
        v > webapp
            > > resources
            v > WEB-INF
                    classes
                > > spring
                v > views
                    v > board
                            boardList.jsp
                            boardView.jsp
                            insertBoard.jsp
                        home.jsp
                        index.jsp
                        login.jsp
                    web.xml
```

- **pom.xml – 라이브러리 추가하기**

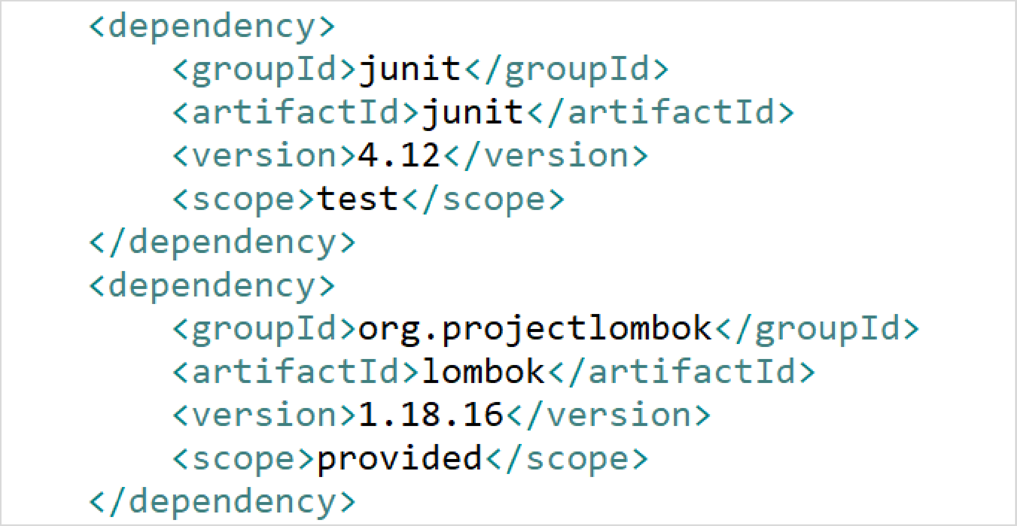


```
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis</artifactId>
    <version>3.5.6</version>
</dependency>
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis-spring</artifactId>
    <version>2.0.6</version>
</dependency>
```

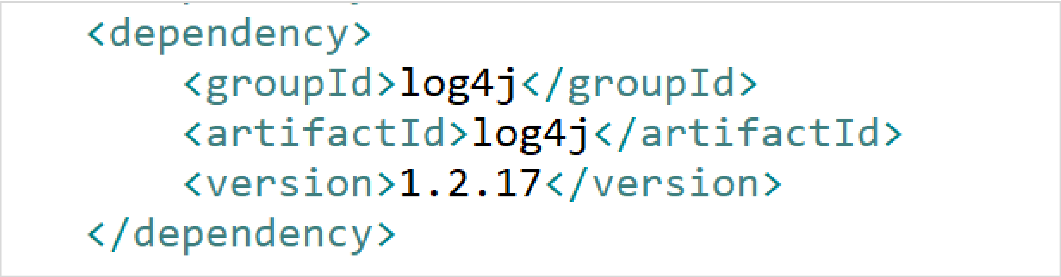


```
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>${org.springframework-version}</version>
</dependency>
<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-tx</artifactId>
    <version>${org.springframework-version}</version>
</dependency>
```

- **pom.xml – 라이브러리 추가하기**

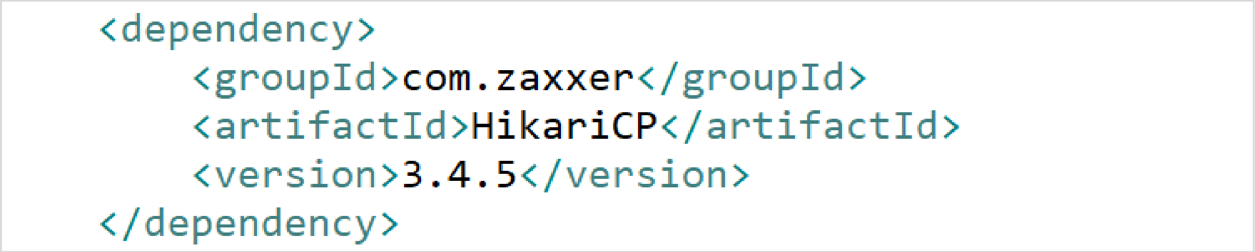


```
    <dependency>
        <groupId>junit</groupId>
        <artifactId>junit</artifactId>
        <version>4.12</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <version>1.18.16</version>
        <scope>provided</scope>
    </dependency>
```

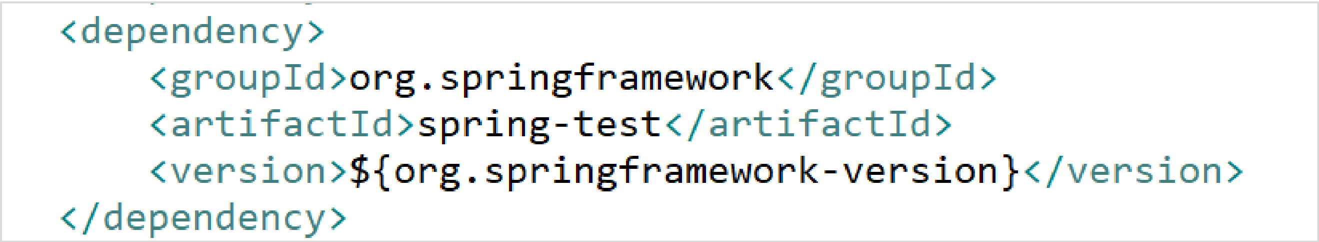


```
    <dependency>
        <groupId>log4j</groupId>
        <artifactId>log4j</artifactId>
        <version>1.2.17</version>
    </dependency>
```

- **pom.xml – 라이브러리 추가하기**

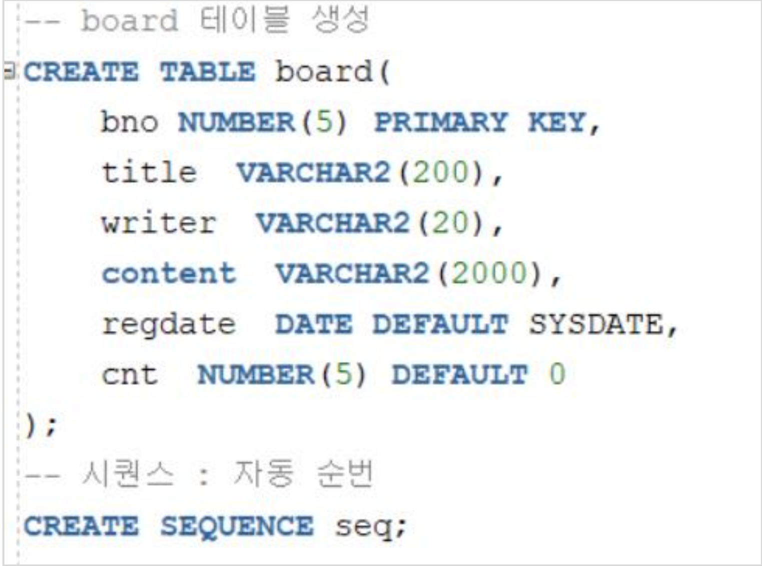


```
    <dependency>
        <groupId>com.zaxxer</groupId>
        <artifactId>HikariCP</artifactId>
        <version>3.4.5</version>
    </dependency>
```

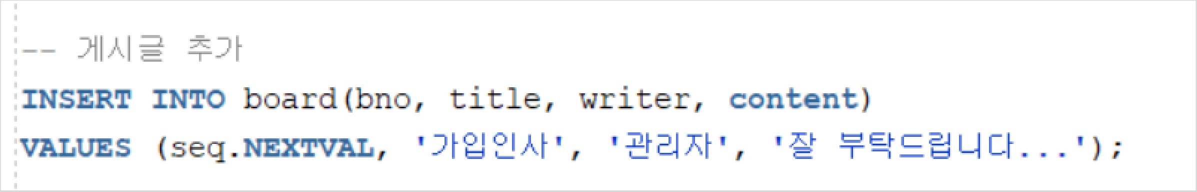


```
  <dependency>
      <groupId>org.springframework</groupId>
      <artifactId>spring-test</artifactId>
      <version>${org.springframework-version}</version>
  </dependency>
```

- **DB 구축**

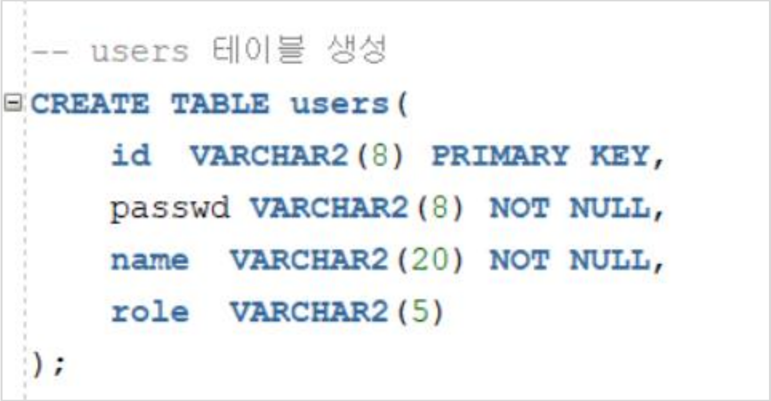


```
-- board 테이블 생성
CREATE TABLE board(
    bno NUMBER(5) PRIMARY KEY,
    title  VARCHAR2(200),
    writer  VARCHAR2(20),
    content  VARCHAR2(2000),
    regdate  DATE DEFAULT SYSDATE,
    cnt  NUMBER(5) DEFAULT 0
);
-- 시퀀스 : 자동 순번
CREATE SEQUENCE seq;
```

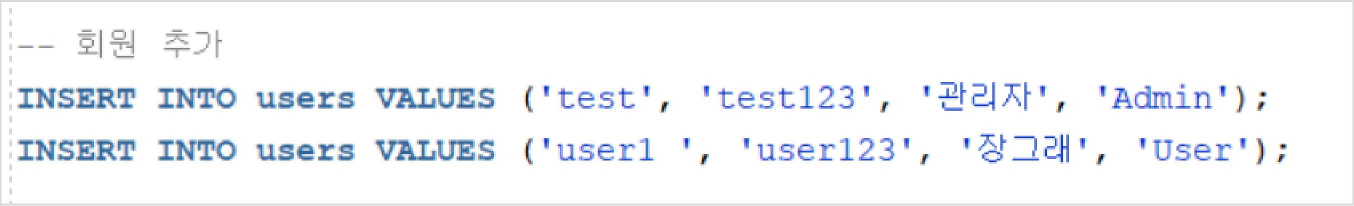


```
-- 게시글 추가
INSERT INTO board(bno, title, writer, content)
VALUES (seq.NEXTVAL, '가입인사', '관리자', '잘 부탁드립니다...');
```

- **DB 구축**

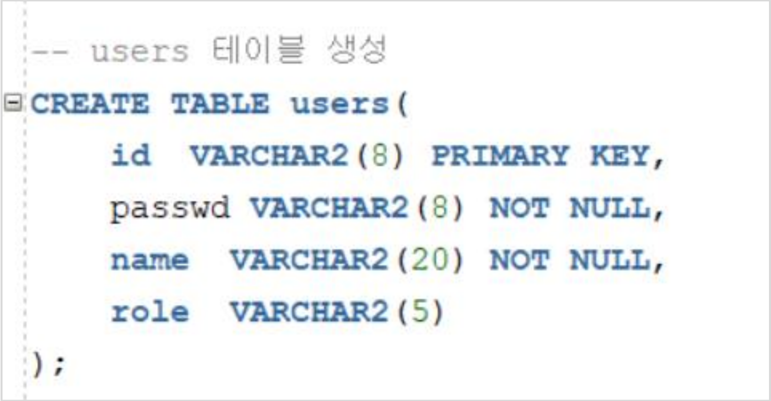


```
-- users 테이블 생성
CREATE TABLE users(
    id  VARCHAR2(8) PRIMARY KEY,
    passwd VARCHAR2(8) NOT NULL,
    name  VARCHAR2(20) NOT NULL,
    role  VARCHAR2(5)
);
```

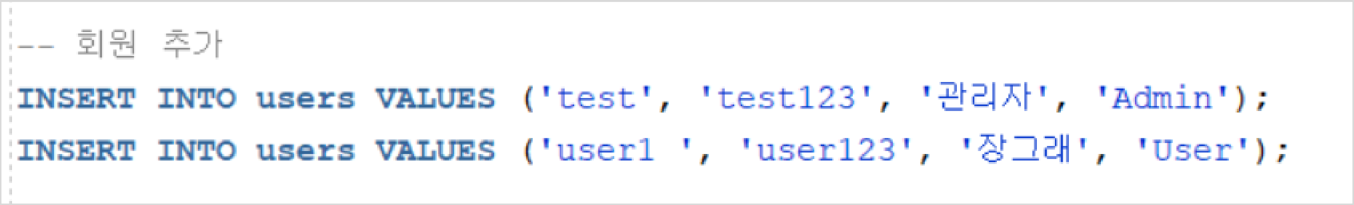


```
-- 회원 추가
INSERT INTO users VALUES ('test', 'test123', '관리자', 'Admin');
INSERT INTO users VALUES ('user1 ', 'user123', '장그래', 'User');
```

- **DB 구축**



```
-- users 테이블 생성
CREATE TABLE users(
    id   VARCHAR2(8)  PRIMARY KEY,
    passwd VARCHAR2(8)  NOT NULL,
    name   VARCHAR2(20) NOT NULL,
    role   VARCHAR2(5)
);
```

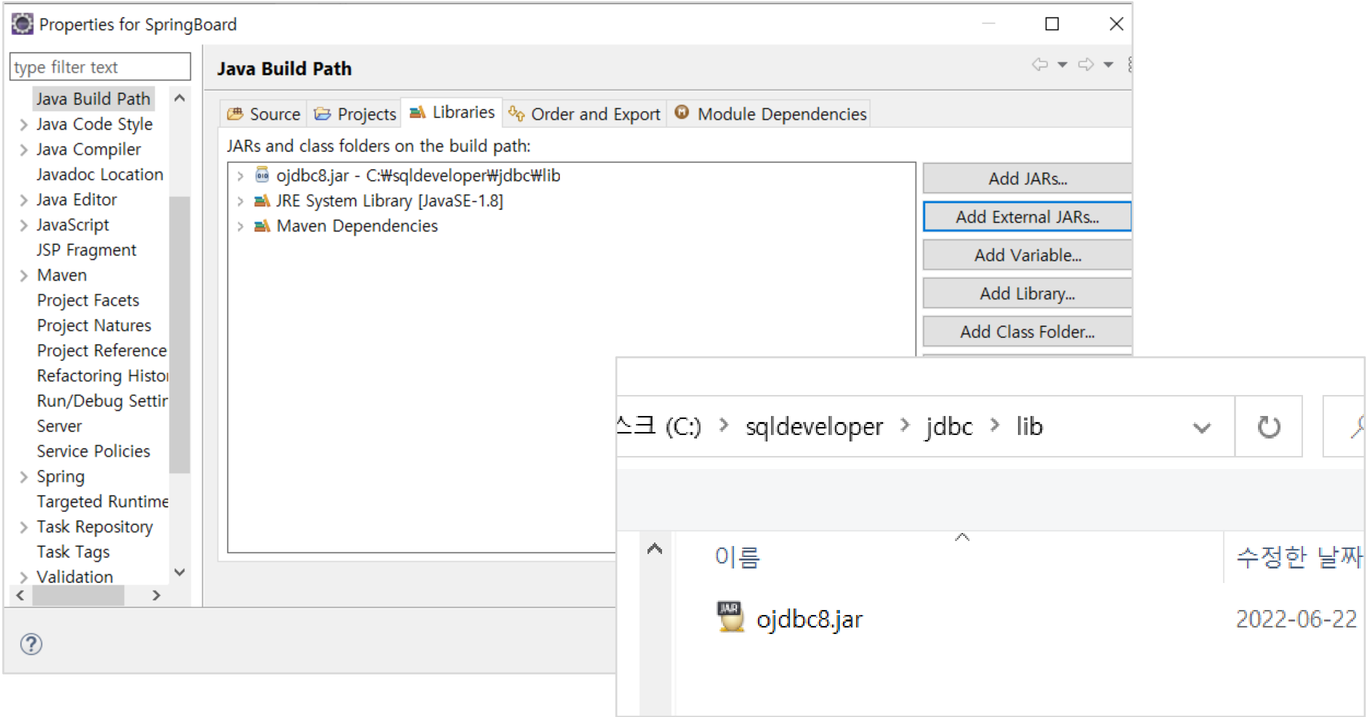


```
-- 회원 추가
INSERT INTO users VALUES ('test', 'test123', '관리자', 'Admin');
INSERT INTO users VALUES ('user1 ', 'user123', '장그래', 'User');
```

오라클 드라이버 연결

- **스프링 프로젝트 – 오라클 드라이버 추가하기**

프로젝트 -> properties -> build path -> Add External Jars -> odjbc8,jar 선택



Properties for SpringBoard

Java Build Path

JARs and class folders on the build path:

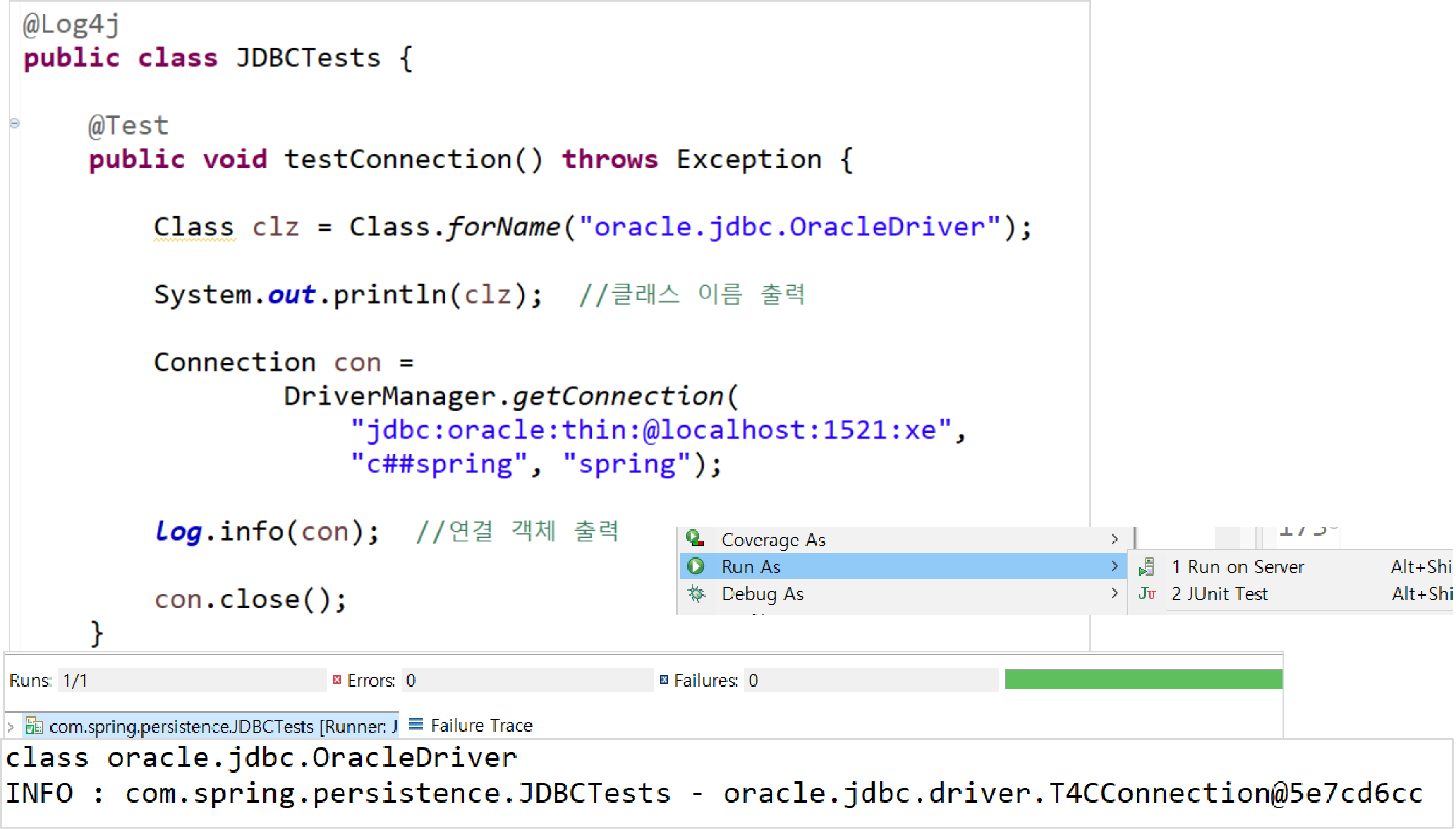
- ojdbc8.jar - C:₩sqldeveloper₩jdbc₩lib
- JRE System Library [JavaSE-1.8]
- Maven Dependencies

스크 (C:) › sqldeveloper › jdbc › lib

이름	수정한 날짜
ojdbc8.jar	2022-06-22

Oracle Database 연동

◆ JDBC 테스트 – Junit



```
@Log4j
public class JDBCTests {

    @Test
    public void testConnection() throws Exception {

        Class clz = Class.forName("oracle.jdbc.OracleDriver");

        System.out.println(clz);  //클래스 이름 출력

        Connection con =
                DriverManager.getConnection(
                    "jdbc:oracle:thin:@localhost:1521:xe",
                    "c##spring", "spring");

        log.info(con);  //연결 객체 출력

        con.close();
    }
```

Runs: 1/1 Errors: 0 Failures: 0

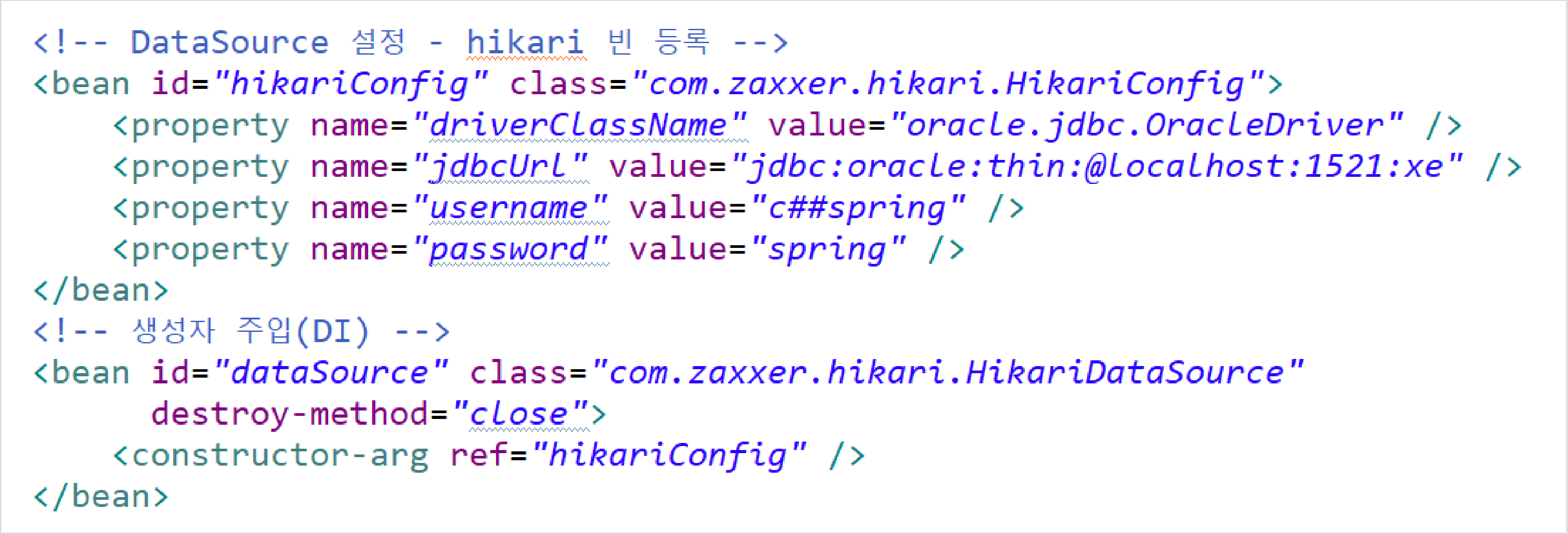
com.spring.persistence.JDBCTests [Runner: J

Failure Trace

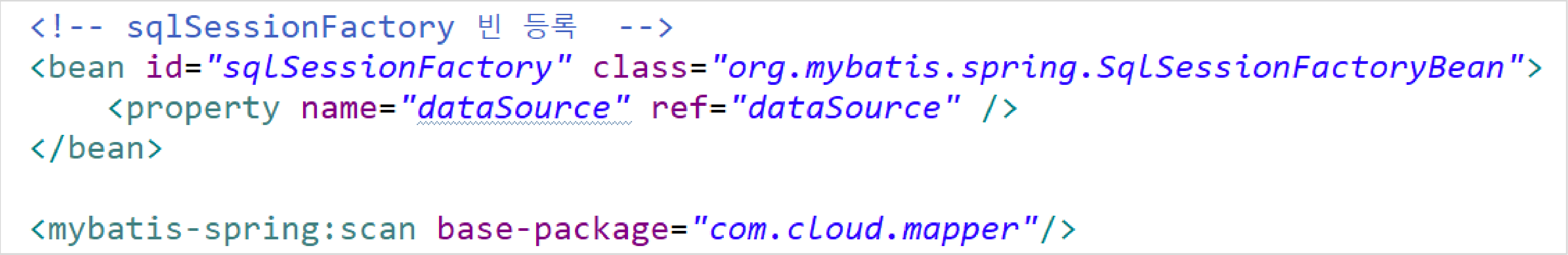
```
class oracle.jdbc.OracleDriver
INFO : com.spring.persistence.JDBCTests - oracle.jdbc.driver.T4CConnection@5e7cd6cc
```

데이터베이스 커넥션 풀(DBCP)

◆ 커넥션 풀(DBCP) 설정 : root-context.xml



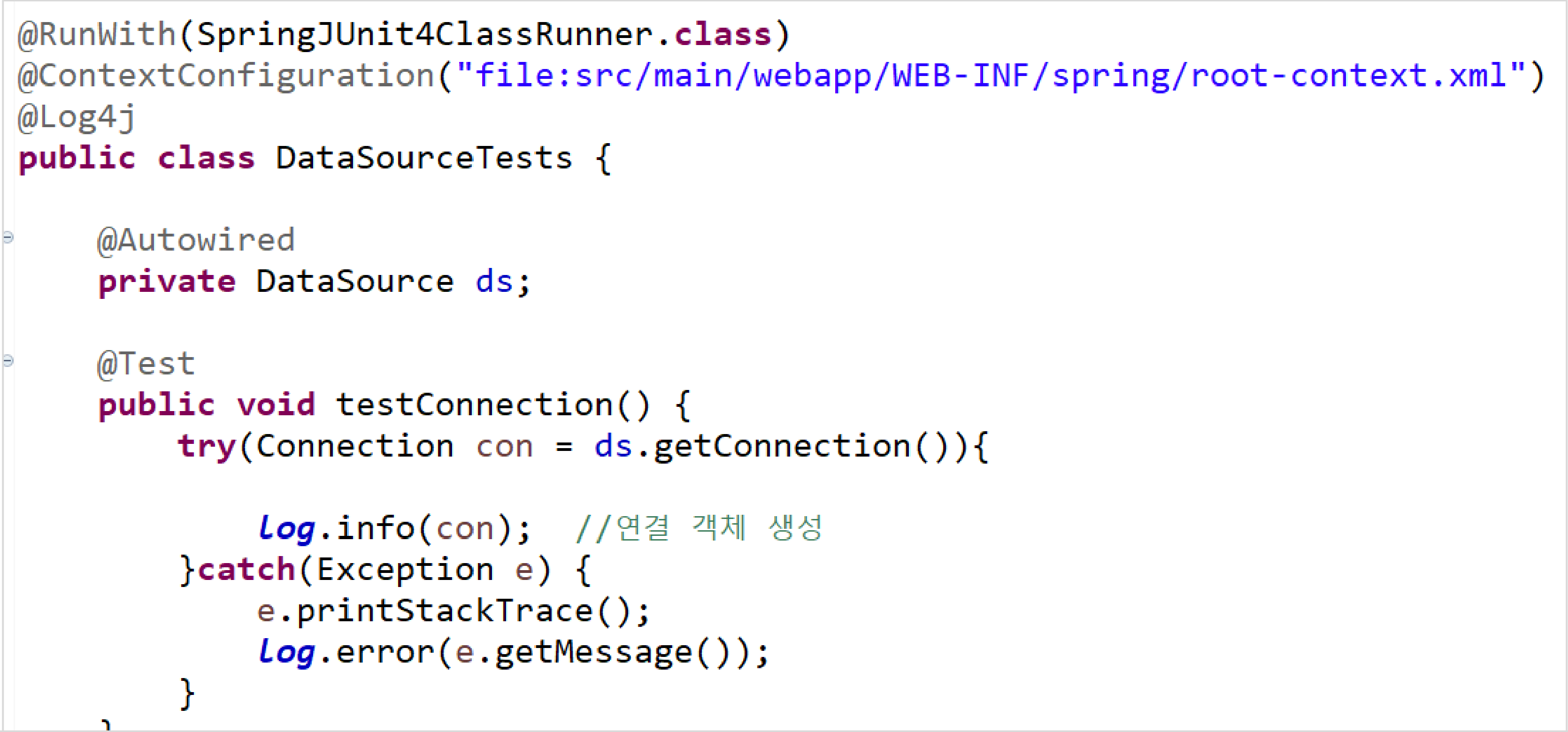
```xml
<!-- DataSource 설정 - hikari 빈 등록 -->
<bean id="hikariConfig" class="com.zaxxer.hikari.HikariConfig">
    <property name="driverClassName" value="oracle.jdbc.OracleDriver" />
    <property name="jdbcUrl" value="jdbc:oracle:thin:@localhost:1521:xe" />
    <property name="username" value="c##spring" />
    <property name="password" value="spring" />
</bean>
<!-- 생성자 주입(DI) -->
<bean id="dataSource" class="com.zaxxer.hikari.HikariDataSource"
      destroy-method="close">
    <constructor-arg ref="hikariConfig" />
</bean>
```



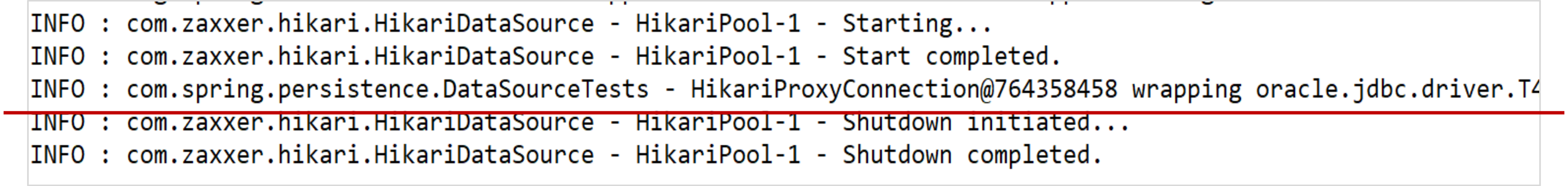
```xml
<!-- sqlSessionFactory 빈 등록  -->
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="dataSource" ref="dataSource" />
</bean>

<mybatis-spring:scan base-package="com.cloud.mapper"/>
```

데이터베이스 커넥션 풀(DBCP)

◆ 커넥션 풀(DBCP) 설정 – Junit



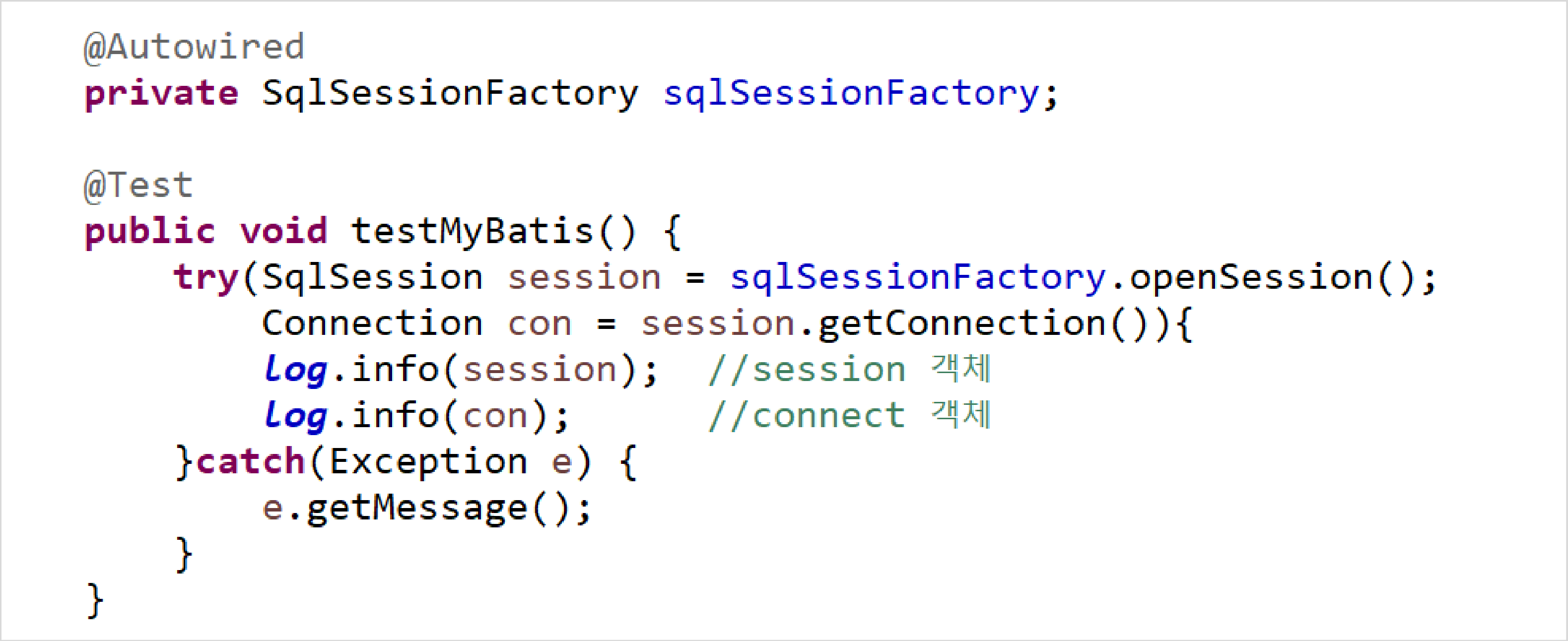
```
@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration("file:src/main/webapp/WEB-INF/spring/root-context.xml")
@Log4j
public class DataSourceTests {

    @Autowired
    private DataSource ds;

    @Test
    public void testConnection() {
        try(Connection con = ds.getConnection()){

            log.info(con);  //연결 객체 생성
        }catch(Exception e) {
            e.printStackTrace();
            log.error(e.getMessage());
        }
```



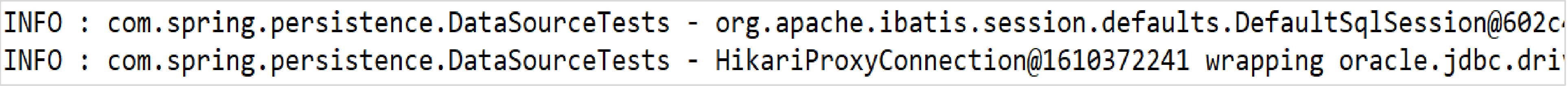
```
INFO : com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Starting...
INFO : com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Start completed.
INFO : com.spring.persistence.DataSourceTests - HikariProxyConnection@764358458 wrapping oracle.jdbc.driver.T4
INFO : com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Shutdown initiated...
INFO : com.zaxxer.hikari.HikariDataSource - HikariPool-1 - Shutdown completed.
```

MyBatis와 스프링 연동

◆ DataSource Test



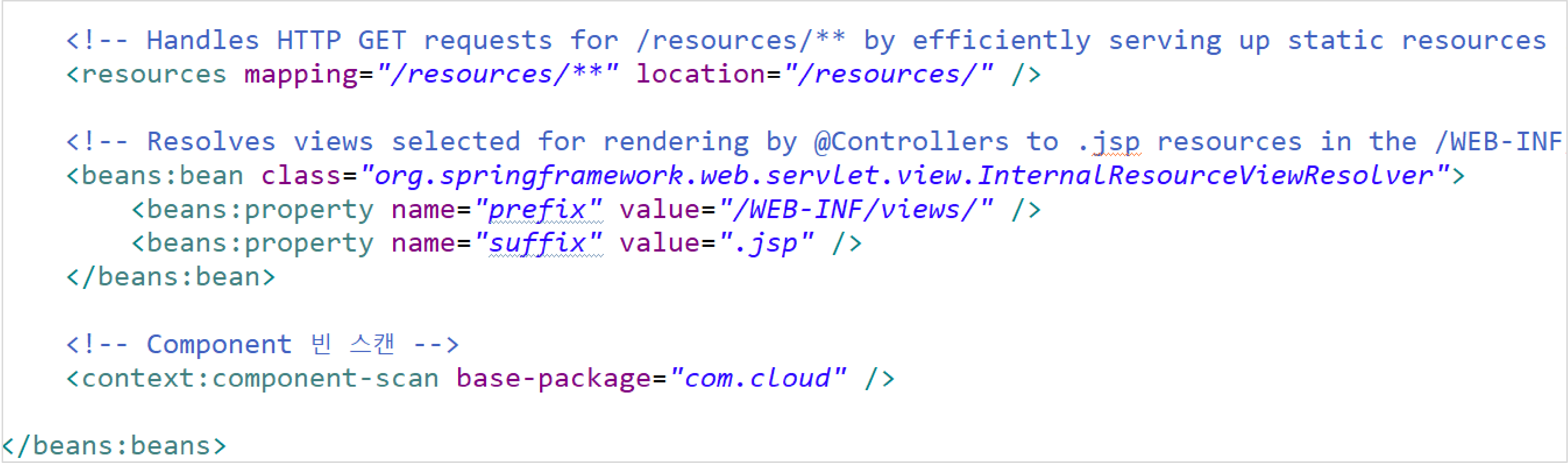
```
@Autowired
private SqlSessionFactory sqlSessionFactory;

@Test
public void testMyBatis() {
    try(SqlSession session = sqlSessionFactory.openSession();
        Connection con = session.getConnection()){
        log.info(session);  //session 객체
        log.info(con);      //connect 객체
    }catch(Exception e) {
        e.getMessage();
    }
}
```



```
INFO : com.spring.persistence.DataSourceTests - org.apache.ibatis.session.defaults.DefaultSqlSession@602c
INFO : com.spring.persistence.DataSourceTests - HikariProxyConnection@1610372241 wrapping oracle.jdbc.dri
```

주요 설정 : servlet-context.xml

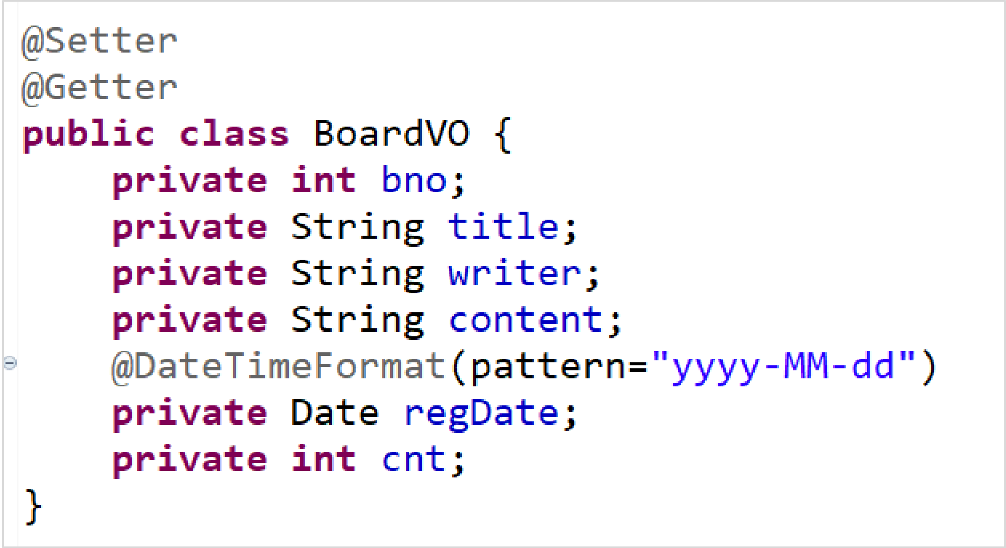
◆ **servlet-context.xml**



```xml
    <!-- Handles HTTP GET requests for /resources/** by efficiently serving up static resources
    <resources mapping="/resources/**" location="/resources/" />

    <!-- Resolves views selected for rendering by @Controllers to .jsp resources in the /WEB-INF
    <beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <beans:property name="prefix" value="/WEB-INF/views/" />
        <beans:property name="suffix" value=".jsp" />
    </beans:bean>

    <!-- Component 빈 스캔 -->
    <context:component-scan base-package="com.cloud" />

</beans:beans>
```

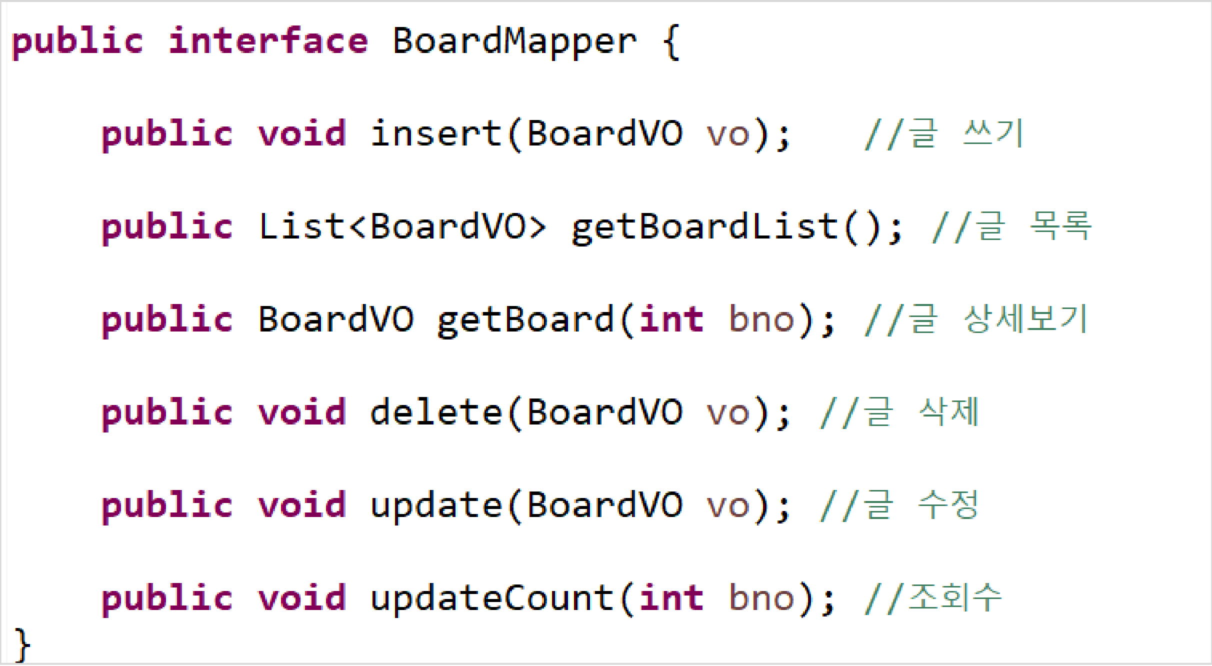
SpringBoard

- VO



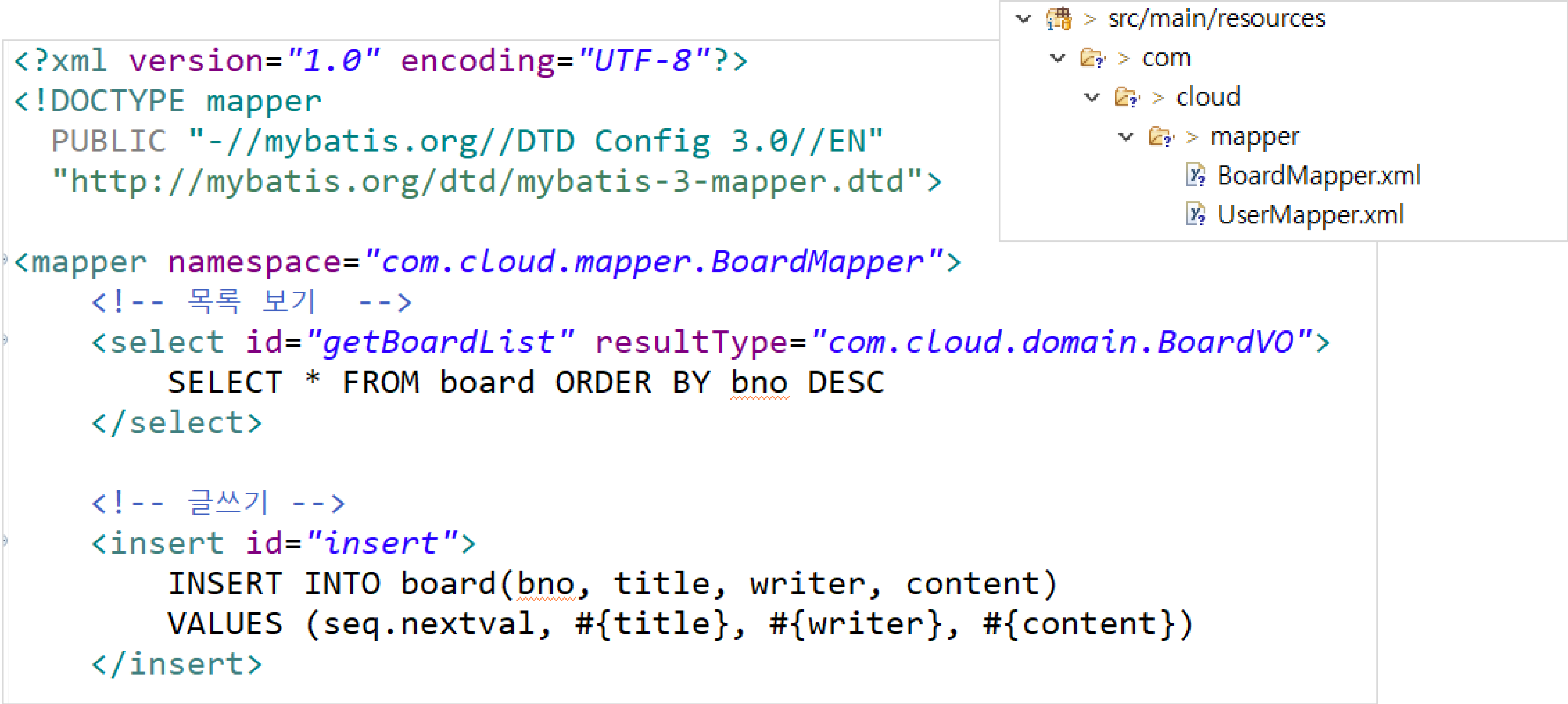
```java
@Setter
@Getter
public class BoardVO {
	private int bno;
	private String title;
	private String writer;
	private String content;
	@DateTimeFormat(pattern="yyyy-MM-dd")
	private Date regDate;
	private int cnt;
}
```

- Mapper 만들기 – BoardMapper.java



```java
public interface BoardMapper {

    public void insert(BoardVO vo);   //글 쓰기

    public List<BoardVO> getBoardList(); //글 목록

    public BoardVO getBoard(int bno); //글 상세보기

    public void delete(BoardVO vo); //글 삭제

    public void update(BoardVO vo); //글 수정

    public void updateCount(int bno); //조회수
}
```

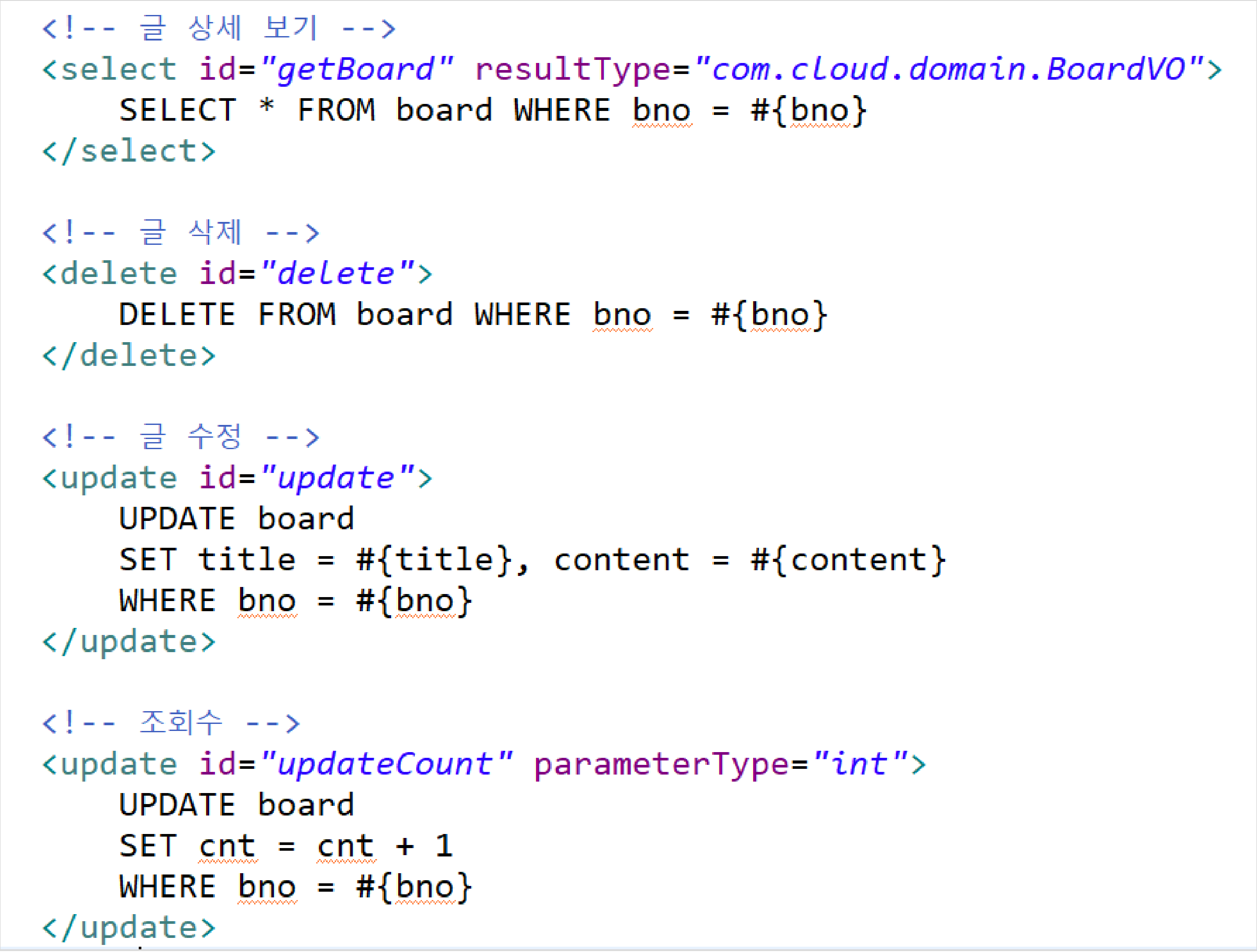
- **BoardMapper.xml**



```
src/main/resources
  com
    cloud
      mapper
        BoardMapper.xml
        UserMapper.xml
```

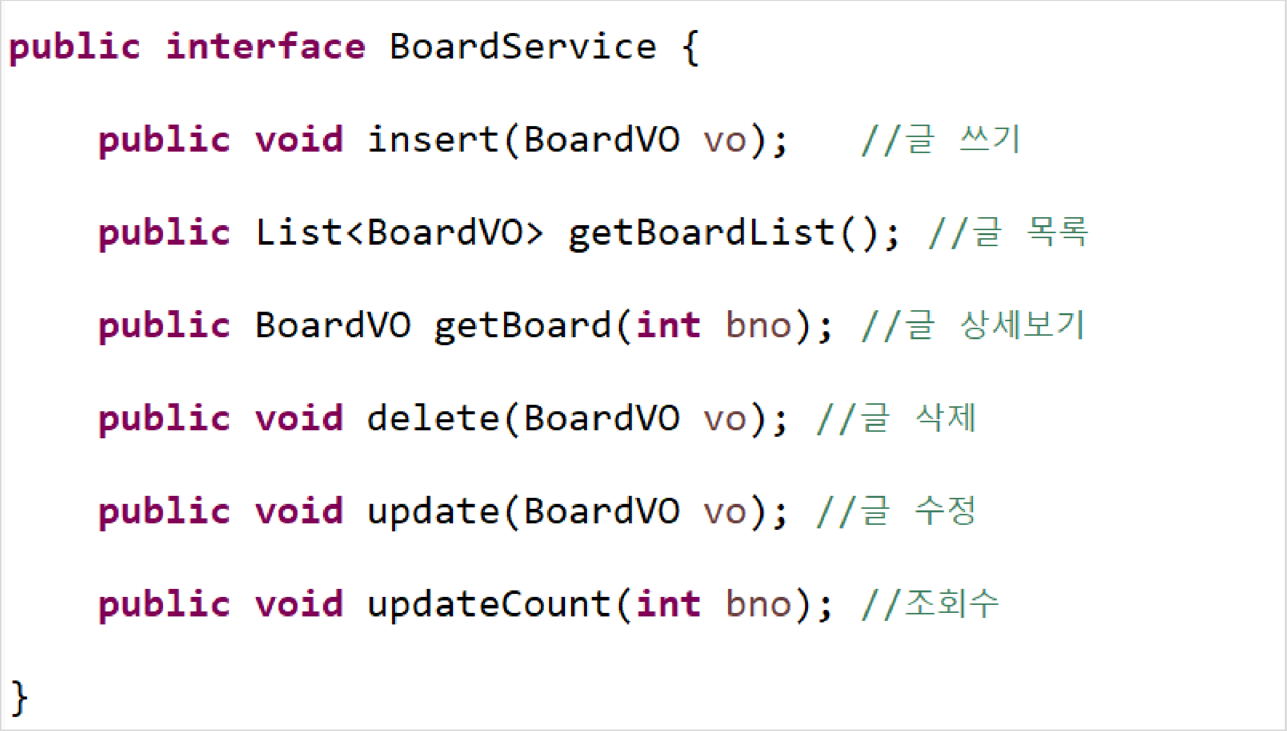
```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.cloud.mapper.BoardMapper">
    <!-- 목록 보기  -->
    <select id="getBoardList" resultType="com.cloud.domain.BoardVO">
        SELECT * FROM board ORDER BY bno DESC
    </select>

    <!-- 글쓰기 -->
    <insert id="insert">
        INSERT INTO board(bno, title, writer, content)
        VALUES (seq.nextval, #{title}, #{writer}, #{content})
    </insert>
```

SpringBoard



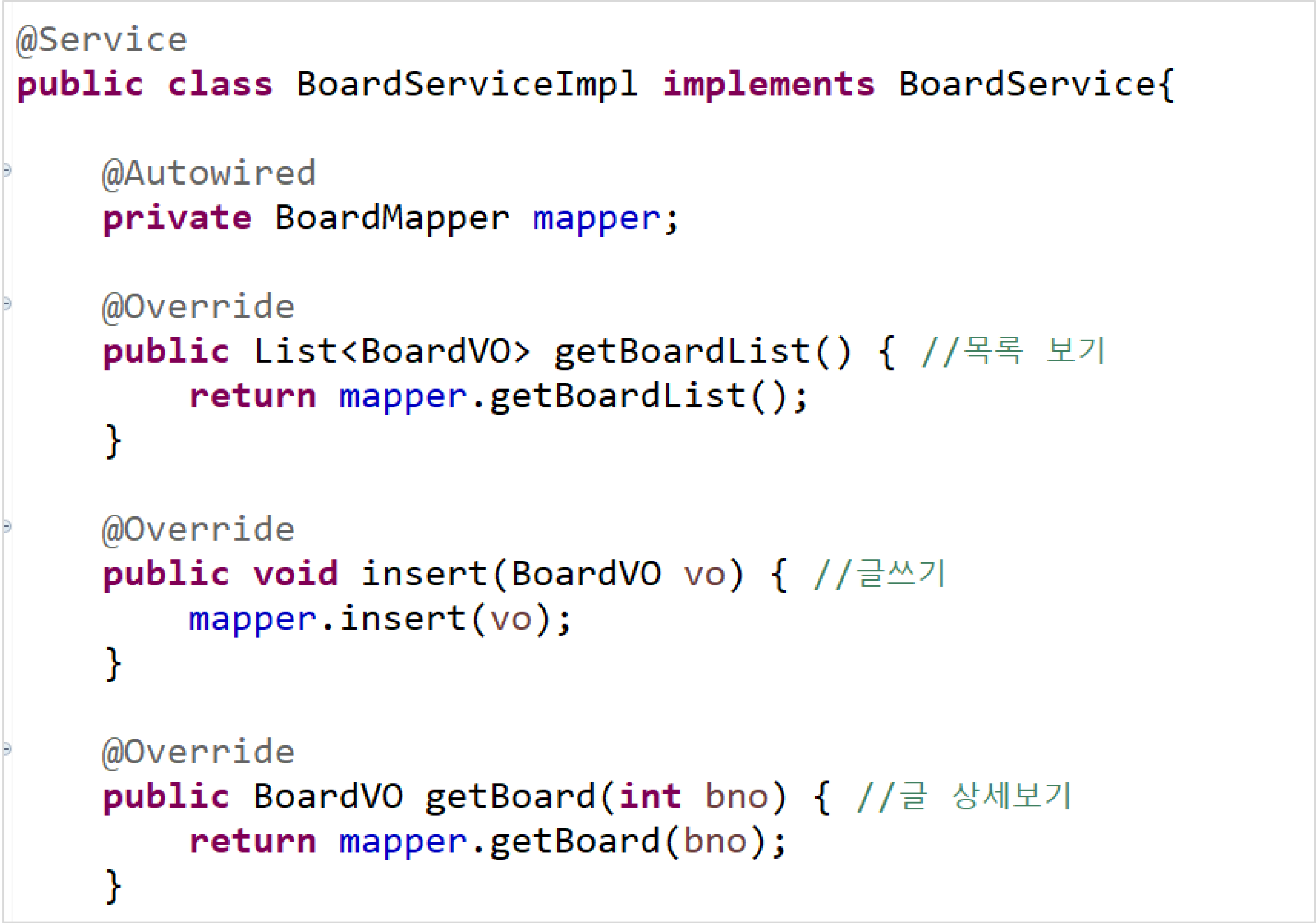
```
<!-- 글 상세 보기 -->
<select id="getBoard" resultType="com.cloud.domain.BoardVO">
    SELECT * FROM board WHERE bno = #{bno}
</select>

<!-- 글 삭제 -->
<delete id="delete">
    DELETE FROM board WHERE bno = #{bno}
</delete>

<!-- 글 수정 -->
<update id="update">
    UPDATE board
    SET title = #{title}, content = #{content}
    WHERE bno = #{bno}
</update>

<!-- 조회수 -->
<update id="updateCount" parameterType="int">
    UPDATE board
    SET cnt = cnt + 1
    WHERE bno = #{bno}
</update>
```

- **Service 계층**



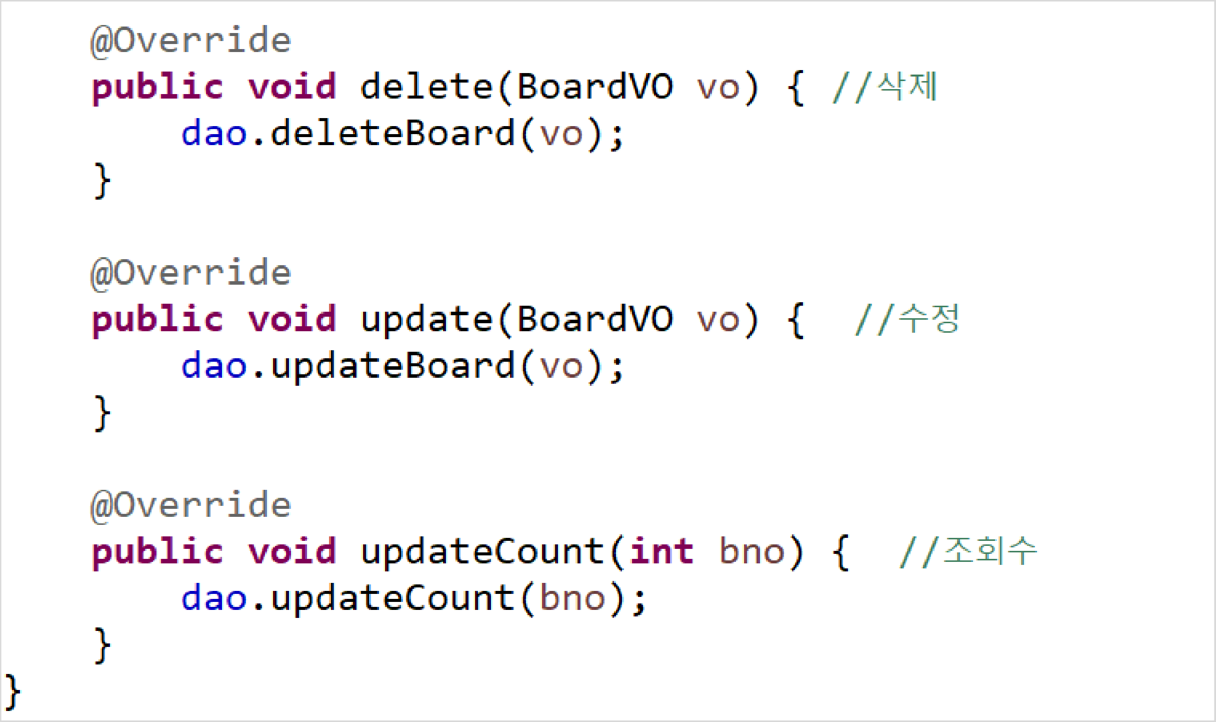
```
public interface BoardService {

    public void insert(BoardVO vo);   //글 쓰기

    public List<BoardVO> getBoardList(); //글 목록

    public BoardVO getBoard(int bno); //글 상세보기

    public void delete(BoardVO vo); //글 삭제

    public void update(BoardVO vo); //글 수정

    public void updateCount(int bno); //조회수

}
```

- **Service 계층**



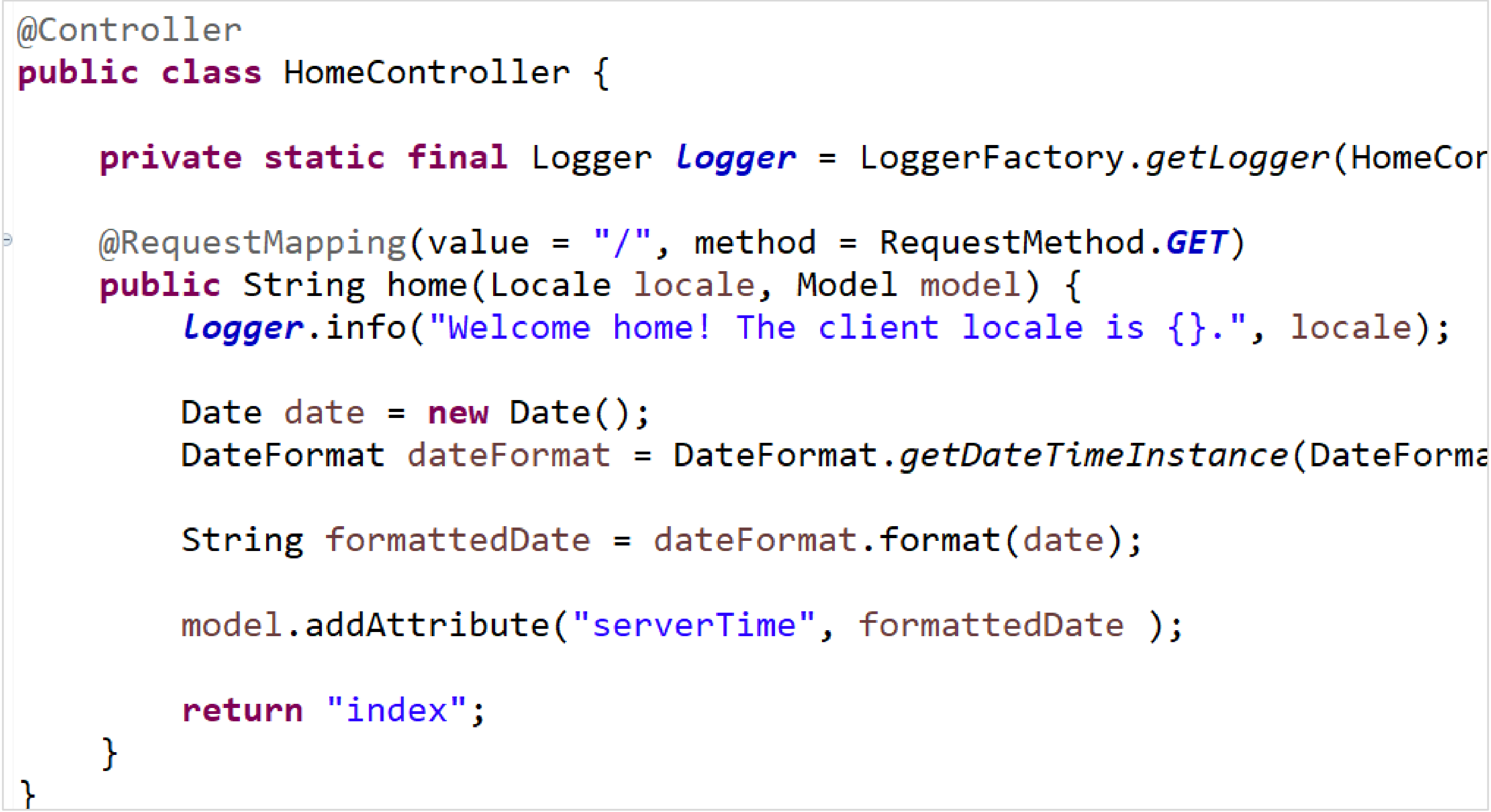
```java
@Service
public class BoardServiceImpl implements BoardService{

    @Autowired
    private BoardMapper mapper;

    @Override
    public List<BoardVO> getBoardList() { //목록 보기
        return mapper.getBoardList();
    }

    @Override
    public void insert(BoardVO vo) { //글쓰기
        mapper.insert(vo);
    }

    @Override
    public BoardVO getBoard(int bno) { //글 상세보기
        return mapper.getBoard(bno);
    }
```

- **Service 계층**



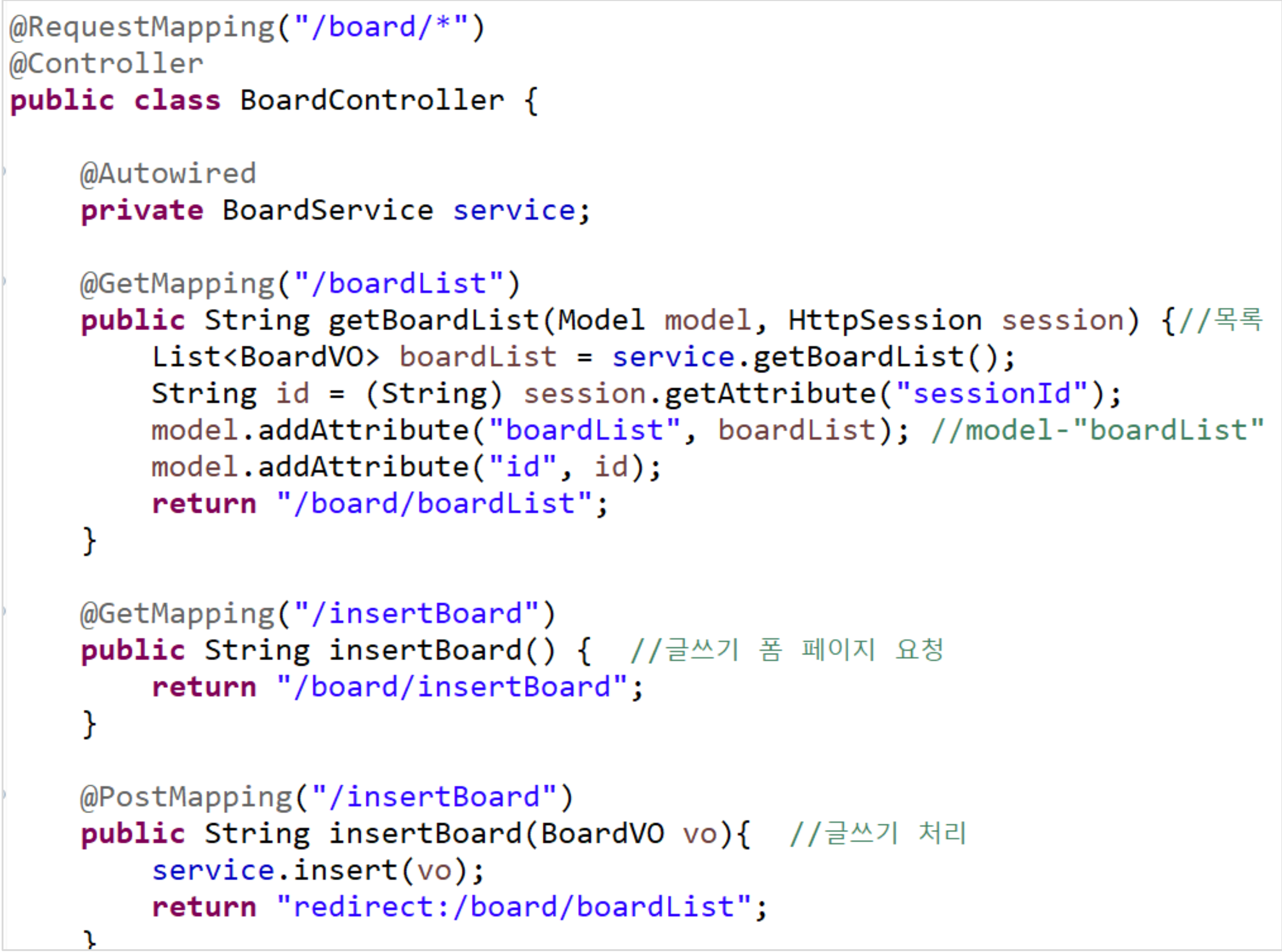
```
    @Override
    public void delete(BoardVO vo) { //삭제
        dao.deleteBoard(vo);
    }

    @Override
    public void update(BoardVO vo) {  //수정
        dao.updateBoard(vo);
    }

    @Override
    public void updateCount(int bno) {  //조회수
        dao.updateCount(bno);
    }
}
```

- **Controller 계층 – HomeController.java**



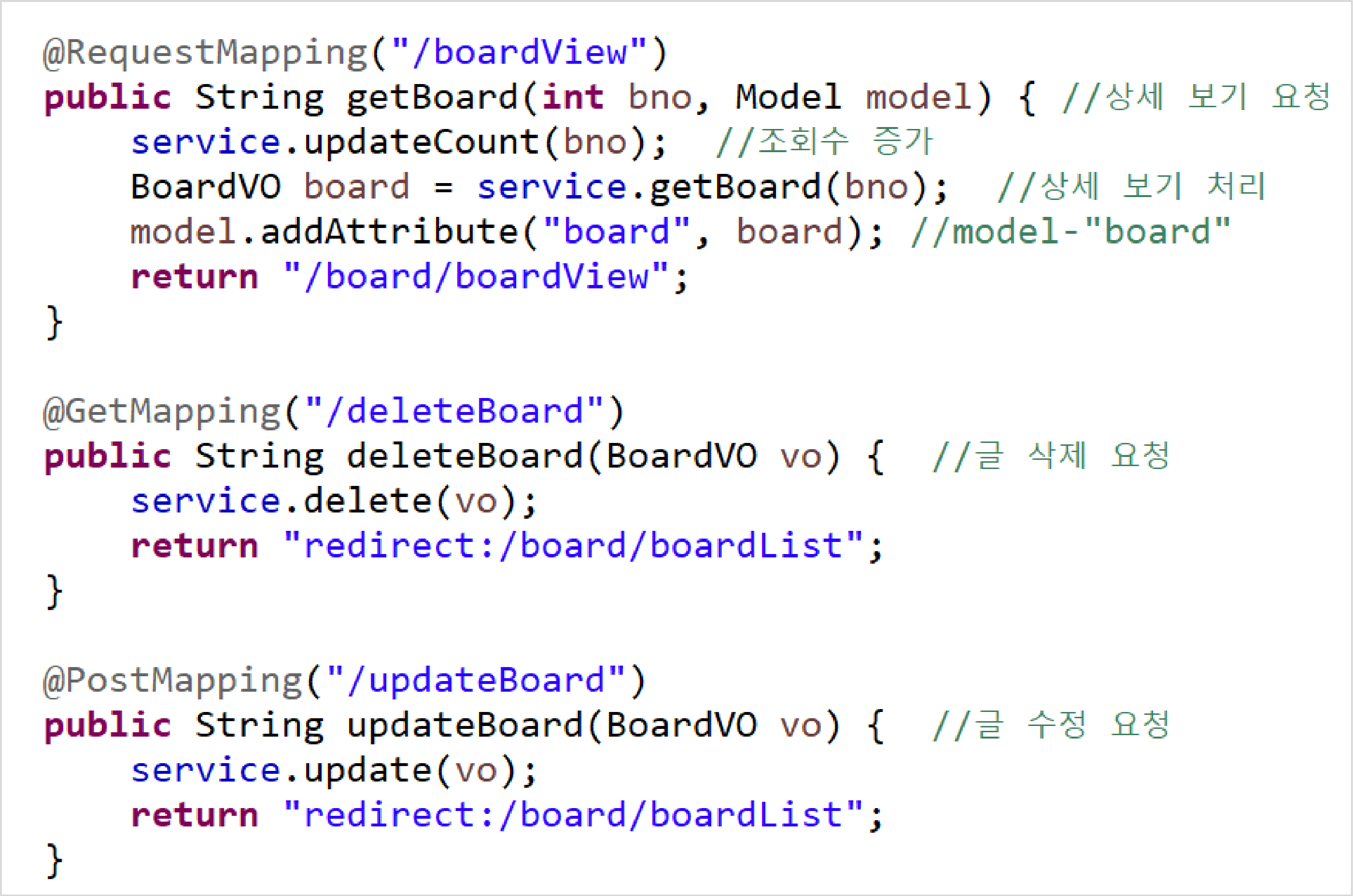
```
@Controller
public class HomeController {

    private static final Logger logger = LoggerFactory.getLogger(HomeCor

    @RequestMapping(value = "/", method = RequestMethod.GET)
    public String home(Locale locale, Model model) {
        logger.info("Welcome home! The client locale is {}.", locale);

        Date date = new Date();
        DateFormat dateFormat = DateFormat.getDateTimeInstance(DateForma

        String formattedDate = dateFormat.format(date);

        model.addAttribute("serverTime", formattedDate );

        return "index";
    }
}
```

- **Controller 계층 – BoardController.java**



```
@RequestMapping("/board/*")
@Controller
public class BoardController {

    @Autowired
    private BoardService service;

    @GetMapping("/boardList")
    public String getBoardList(Model model, HttpSession session) {//목록
        List<BoardVO> boardList = service.getBoardList();
        String id = (String) session.getAttribute("sessionId");
        model.addAttribute("boardList", boardList); //model-"boardList"
        model.addAttribute("id", id);
        return "/board/boardList";
    }

    @GetMapping("/insertBoard")
    public String insertBoard() {  //글쓰기 폼 페이지 요청
        return "/board/insertBoard";
    }

    @PostMapping("/insertBoard")
    public String insertBoard(BoardVO vo){  //글쓰기 처리
        service.insert(vo);
        return "redirect:/board/boardList";
    }
```

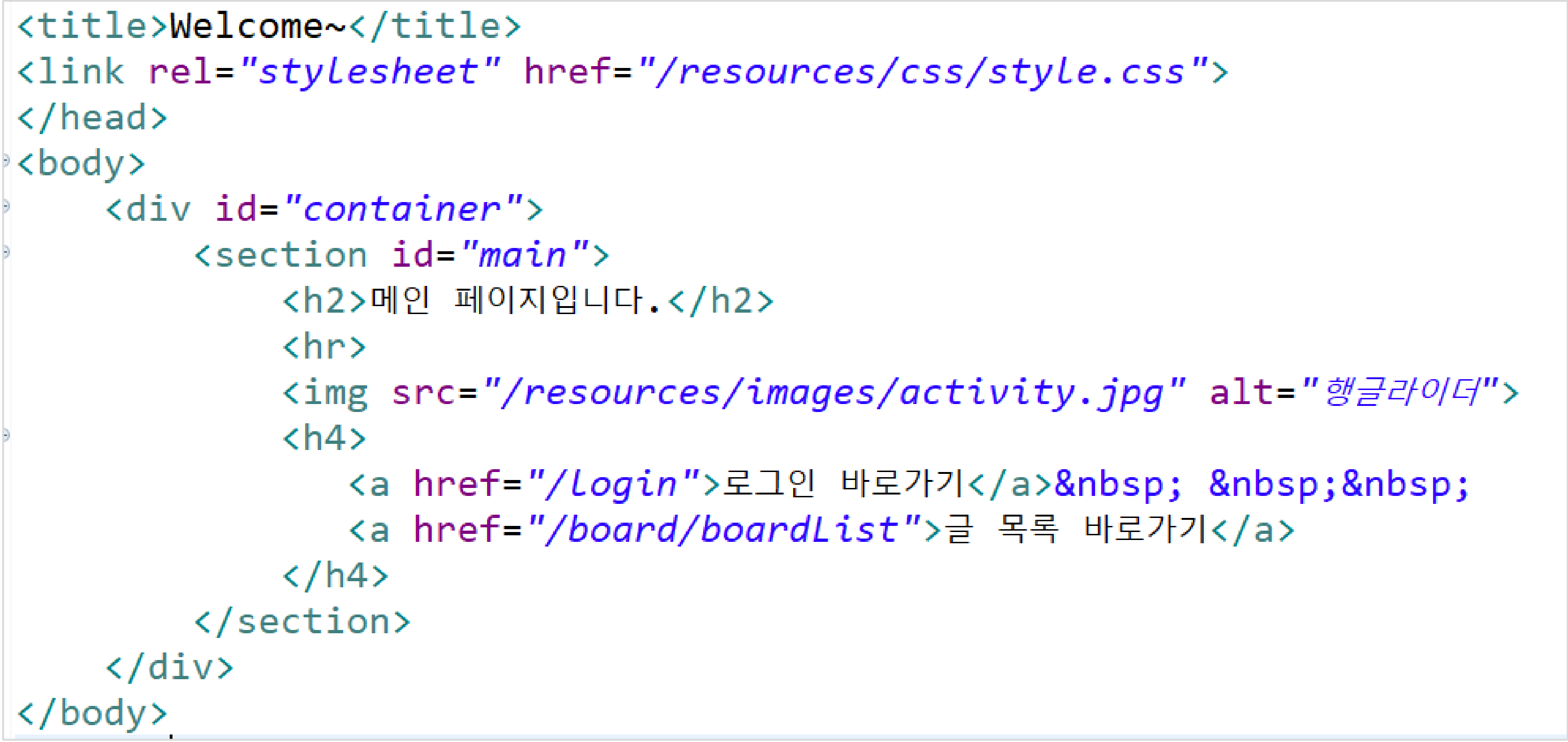
SpringBoard



```
@RequestMapping("/boardView")
public String getBoard(int bno, Model model) { //상세 보기 요청
    service.updateCount(bno);  //조회수 증가
    BoardVO board = service.getBoard(bno);  //상세 보기 처리
    model.addAttribute("board", board); //model-"board"
    return "/board/boardView";
}

@GetMapping("/deleteBoard")
public String deleteBoard(BoardVO vo) {  //글 삭제 요청
    service.delete(vo);
    return "redirect:/board/boardList";
}

@PostMapping("/updateBoard")
public String updateBoard(BoardVO vo) {  //글 수정 요청
    service.update(vo);
    return "redirect:/board/boardList";
}
```

SpringBoard

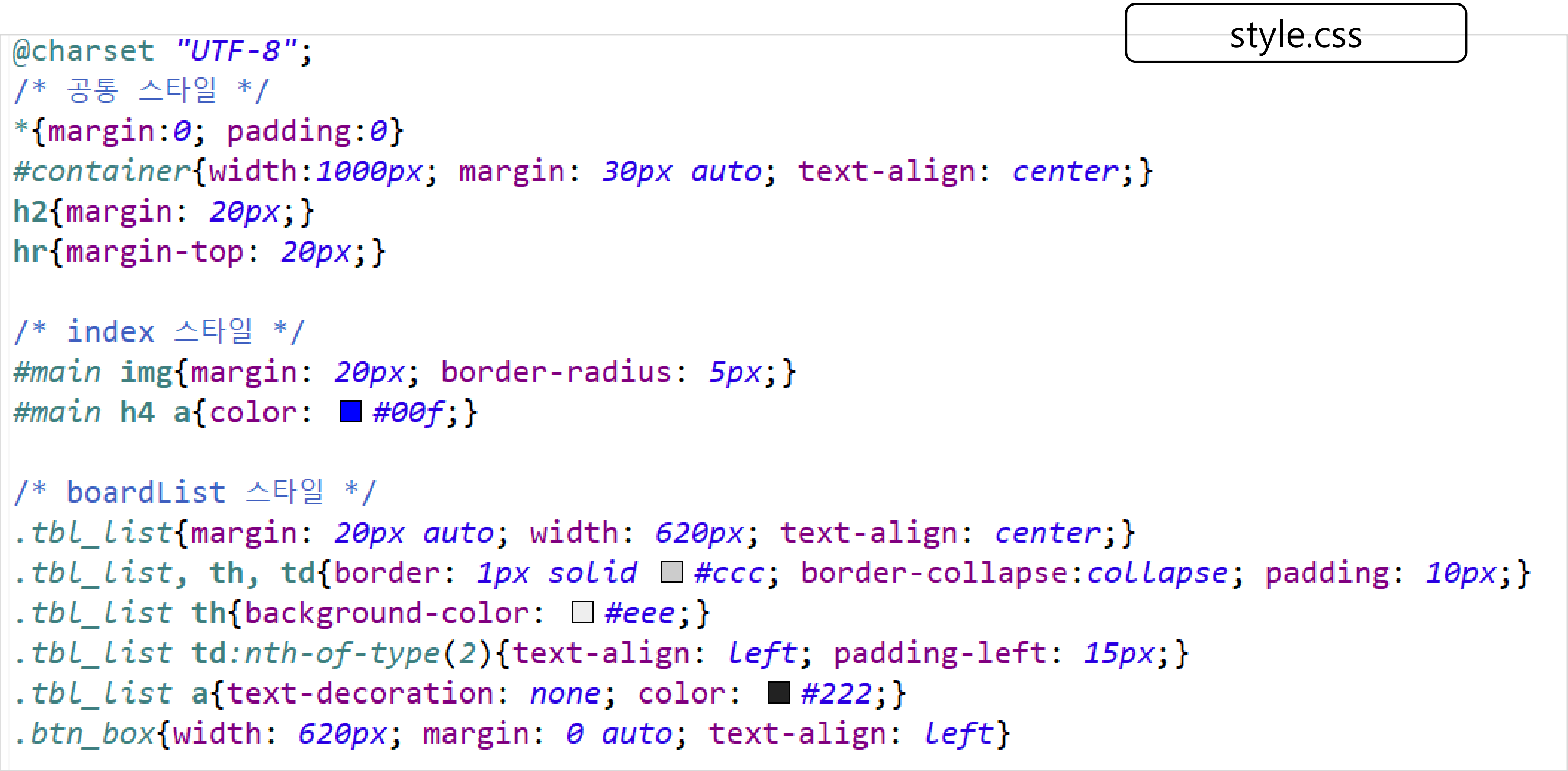
index.jsp



```
<title>Welcome~</title>
<link rel="stylesheet" href="/resources/css/style.css">
</head>
<body>
    <div id="container">
        <section id="main">
            <h2>메인 페이지입니다.</h2>
            <hr>
            <img src="/resources/images/activity.jpg" alt="행글라이더">
            <h4>
               <a href="/login">로그인 바로가기</a>&nbsp; &nbsp;&nbsp;
               <a href="/board/boardList">글 목록 바로가기</a>
            </h4>
        </section>
    </div>
</body>
```

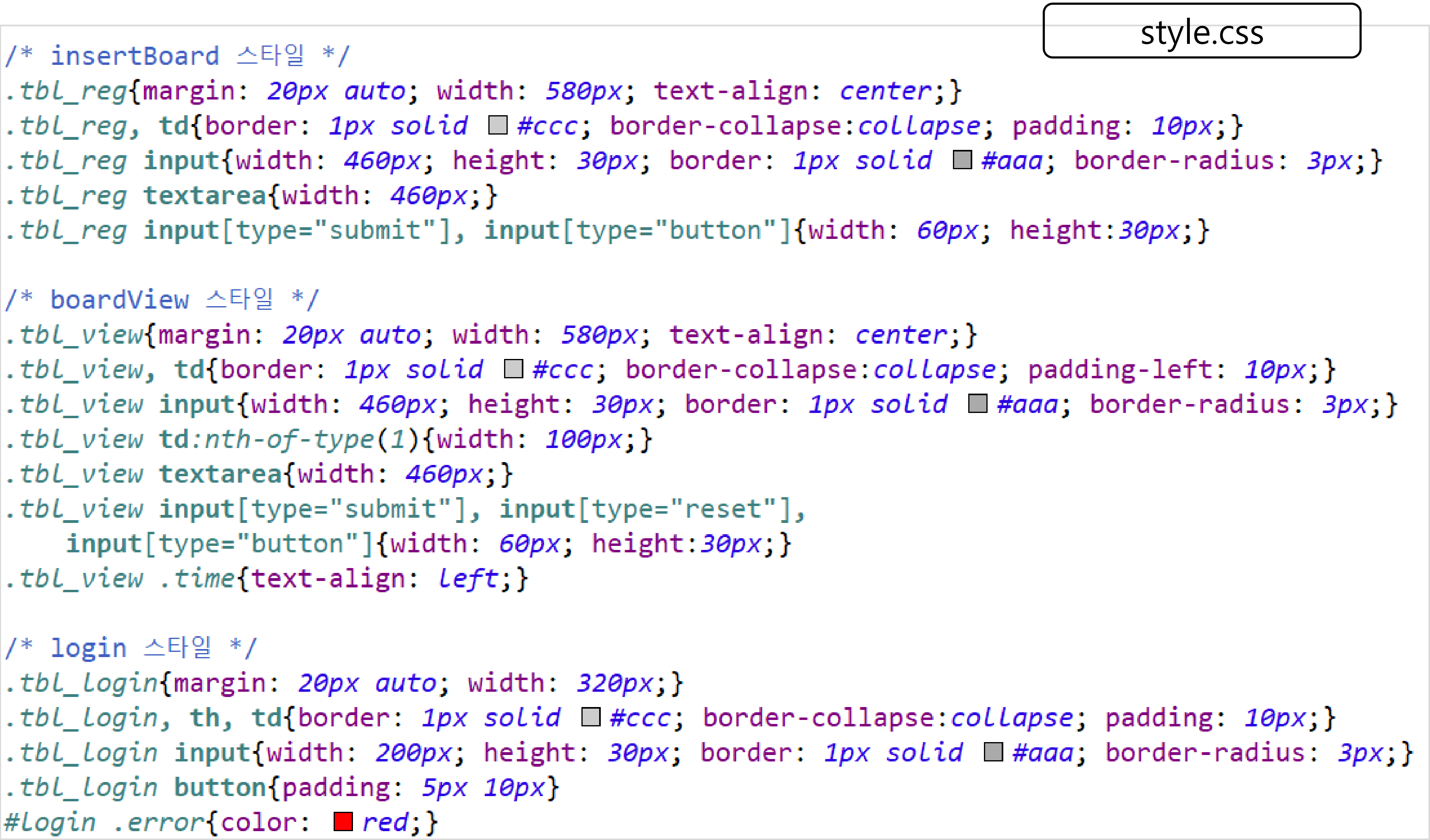
SpringBoard

style.css

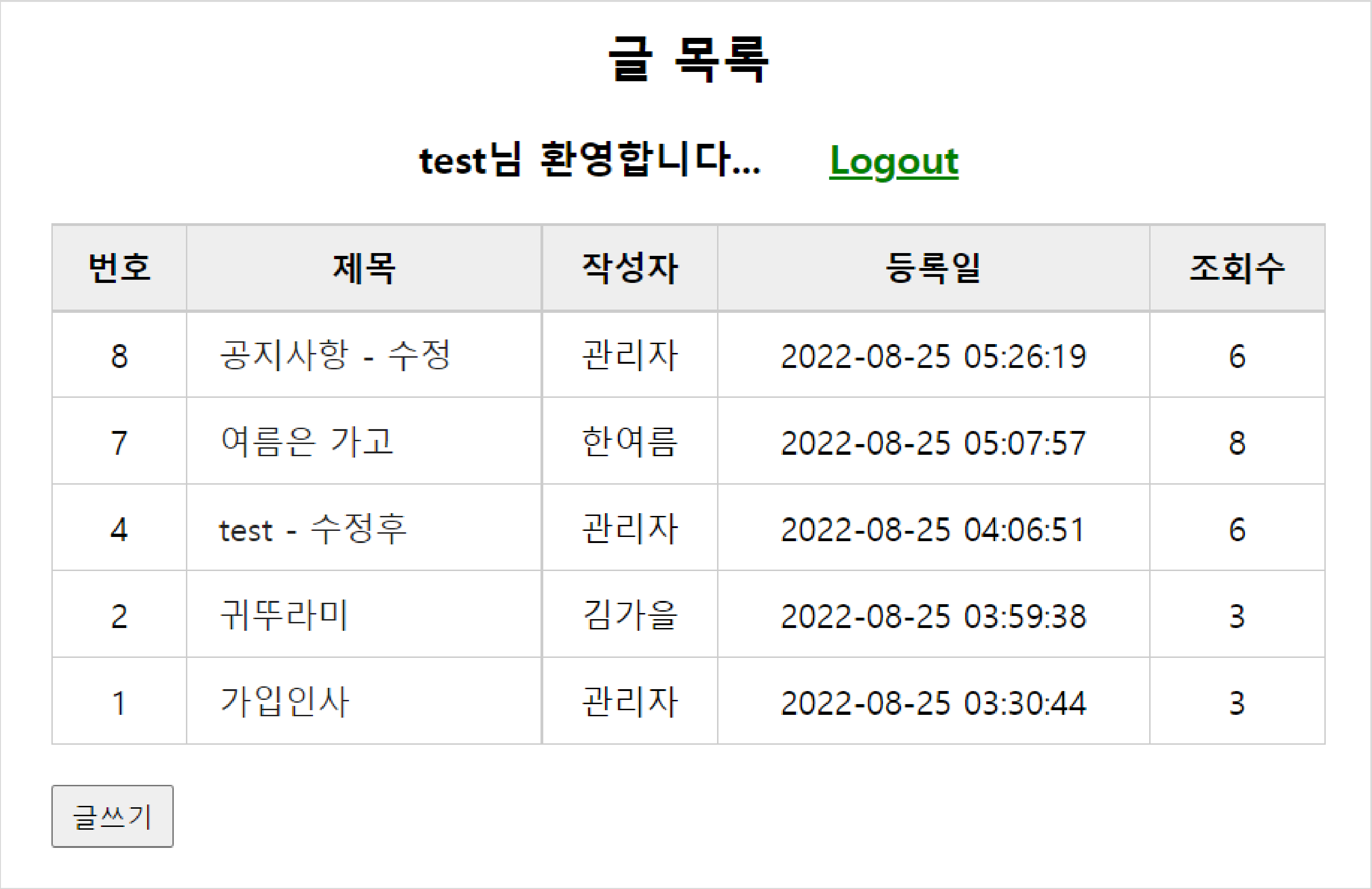


```css
@charset "UTF-8";
/* 공통 스타일 */
*{margin:0; padding:0}
#container{width:1000px; margin: 30px auto; text-align: center;}
h2{margin: 20px;}
hr{margin-top: 20px;}

/* index 스타일 */
#main img{margin: 20px; border-radius: 5px;}
#main h4 a{color: ■ #00f;}

/* boardList 스타일 */
.tbl_list{margin: 20px auto; width: 620px; text-align: center;}
.tbl_list, th, td{border: 1px solid ■ #ccc; border-collapse:collapse; padding: 10px;}
.tbl_list th{background-color: □ #eee;}
.tbl_list td:nth-of-type(2){text-align: left; padding-left: 15px;}
.tbl_list a{text-decoration: none; color: ■ #222;}
.btn_box{width: 620px; margin: 0 auto; text-align: left}
```

style.css



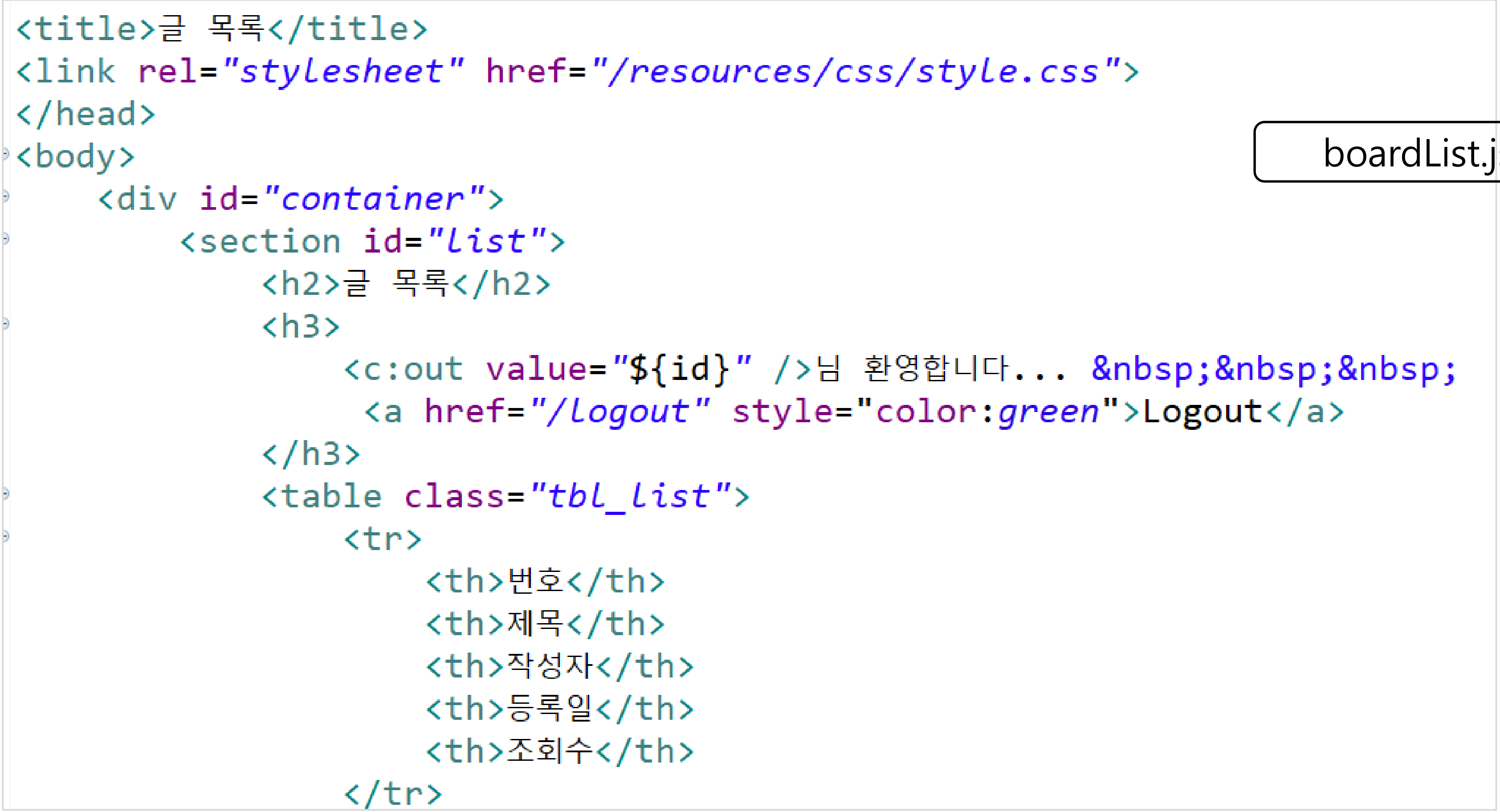
```css
/* insertBoard 스타일 */
.tbl_reg{margin: 20px auto; width: 580px; text-align: center;}
.tbl_reg, td{border: 1px solid #ccc; border-collapse:collapse; padding: 10px;}
.tbl_reg input{width: 460px; height: 30px; border: 1px solid #aaa; border-radius: 3px;}
.tbl_reg textarea{width: 460px;}
.tbl_reg input[type="submit"], input[type="button"]{width: 60px; height:30px;}

/* boardView 스타일 */
.tbl_view{margin: 20px auto; width: 580px; text-align: center;}
.tbl_view, td{border: 1px solid #ccc; border-collapse:collapse; padding-left: 10px;}
.tbl_view input{width: 460px; height: 30px; border: 1px solid #aaa; border-radius: 3px;}
.tbl_view td:nth-of-type(1){width: 100px;}
.tbl_view textarea{width: 460px;}
.tbl_view input[type="submit"], input[type="reset"],
    input[type="button"]{width: 60px; height:30px;}
.tbl_view .time{text-align: left;}

/* login 스타일 */
.tbl_login{margin: 20px auto; width: 320px;}
.tbl_login, th, td{border: 1px solid #ccc; border-collapse:collapse; padding: 10px;}
.tbl_login input{width: 200px; height: 30px; border: 1px solid #aaa; border-radius: 3px;}
.tbl_login button{padding: 5px 10px}
#login .error{color: red;}
```



글 목록

test님 환영합니다... Logout

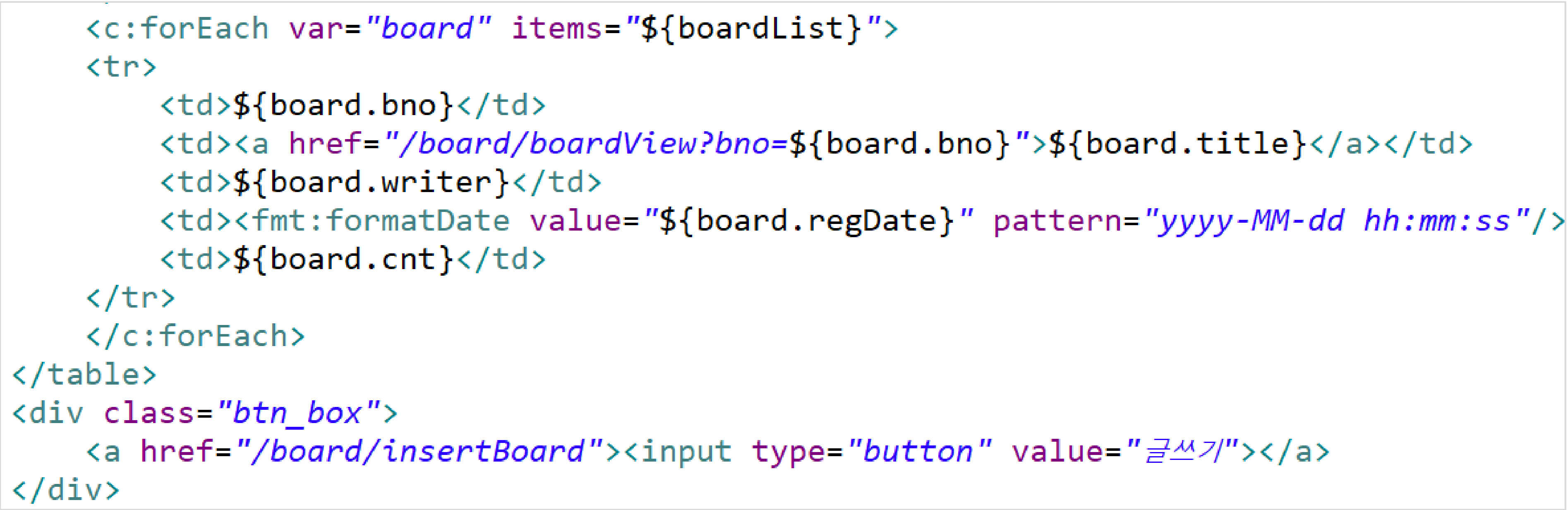
번호	제목	작성자	등록일	조회수
8	공지사항 - 수정	관리자	2022-08-25 05:26:19	6
7	여름은 가고	한여름	2022-08-25 05:07:57	8
4	test - 수정후	관리자	2022-08-25 04:06:51	6
2	귀뚜라미	김가을	2022-08-25 03:59:38	3
1	가입인사	관리자	2022-08-25 03:30:44	3



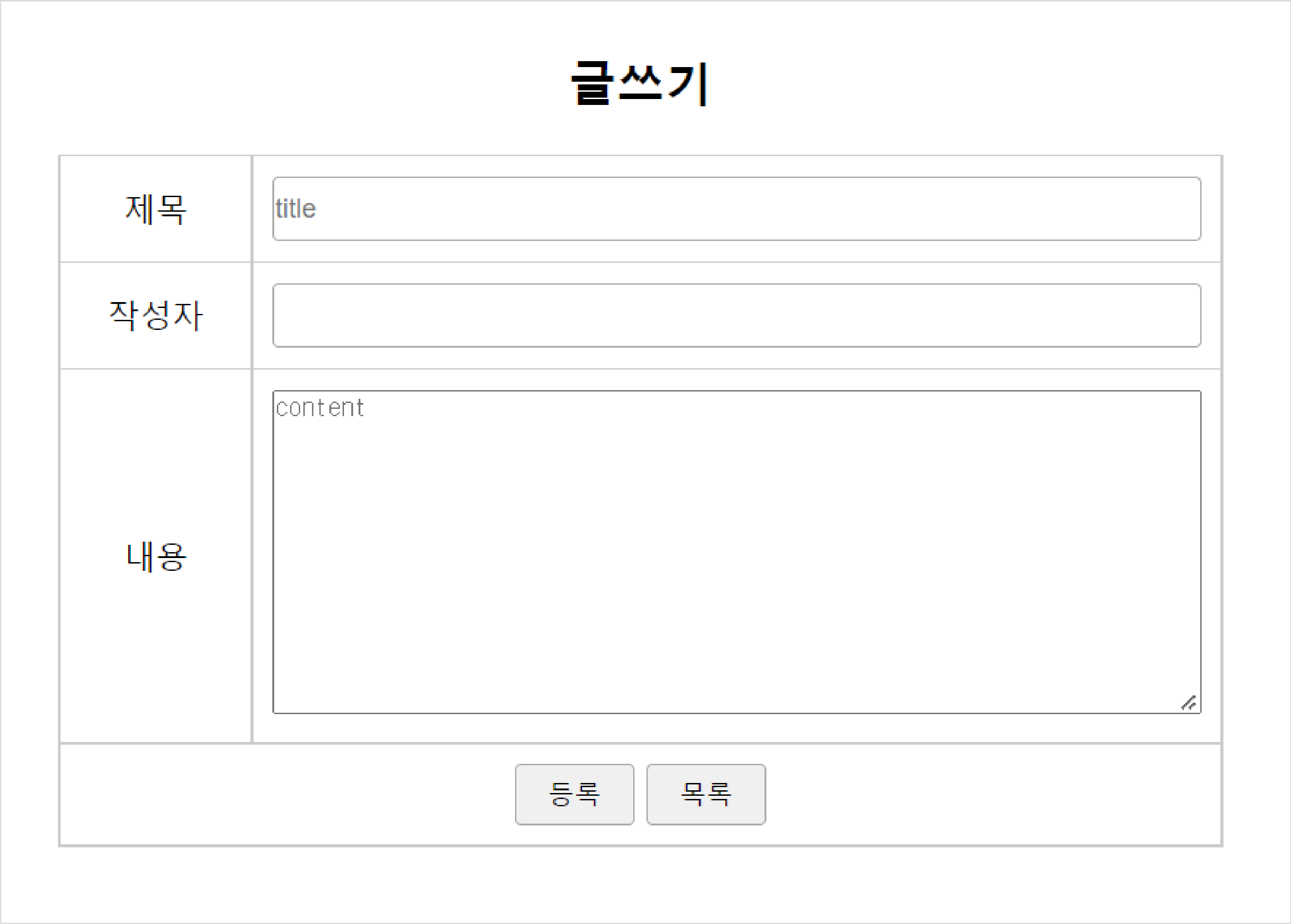
```
<title>글 목록</title>
<link rel="stylesheet" href="/resources/css/style.css">
</head>
<body>
    <div id="container">
        <section id="list">
            <h2>글 목록</h2>
            <h3>
                <c:out value="${id}" />님 환영합니다... &nbsp;&nbsp;&nbsp;
                 <a href="/logout" style="color:green">Logout</a>
            </h3>
            <table class="tbl_list">
                <tr>
                    <th>번호</th>
                    <th>제목</th>
                    <th>작성자</th>
                    <th>등록일</th>
                    <th>조회수</th>
                </tr>
```

boardList.jsp

boardList.jsp



```
    <c:forEach var="board" items="${boardList}">
    <tr>
        <td>${board.bno}</td>
        <td><a href="/board/boardView?bno=${board.bno}">${board.title}</a></td>
        <td>${board.writer}</td>
        <td><fmt:formatDate value="${board.regDate}" pattern="yyyy-MM-dd hh:mm:ss"/>
        <td>${board.cnt}</td>
    </tr>
    </c:forEach>
</table>
<div class="btn_box">
    <a href="/board/insertBoard"><input type="button" value="글쓰기"></a>
</div>
```

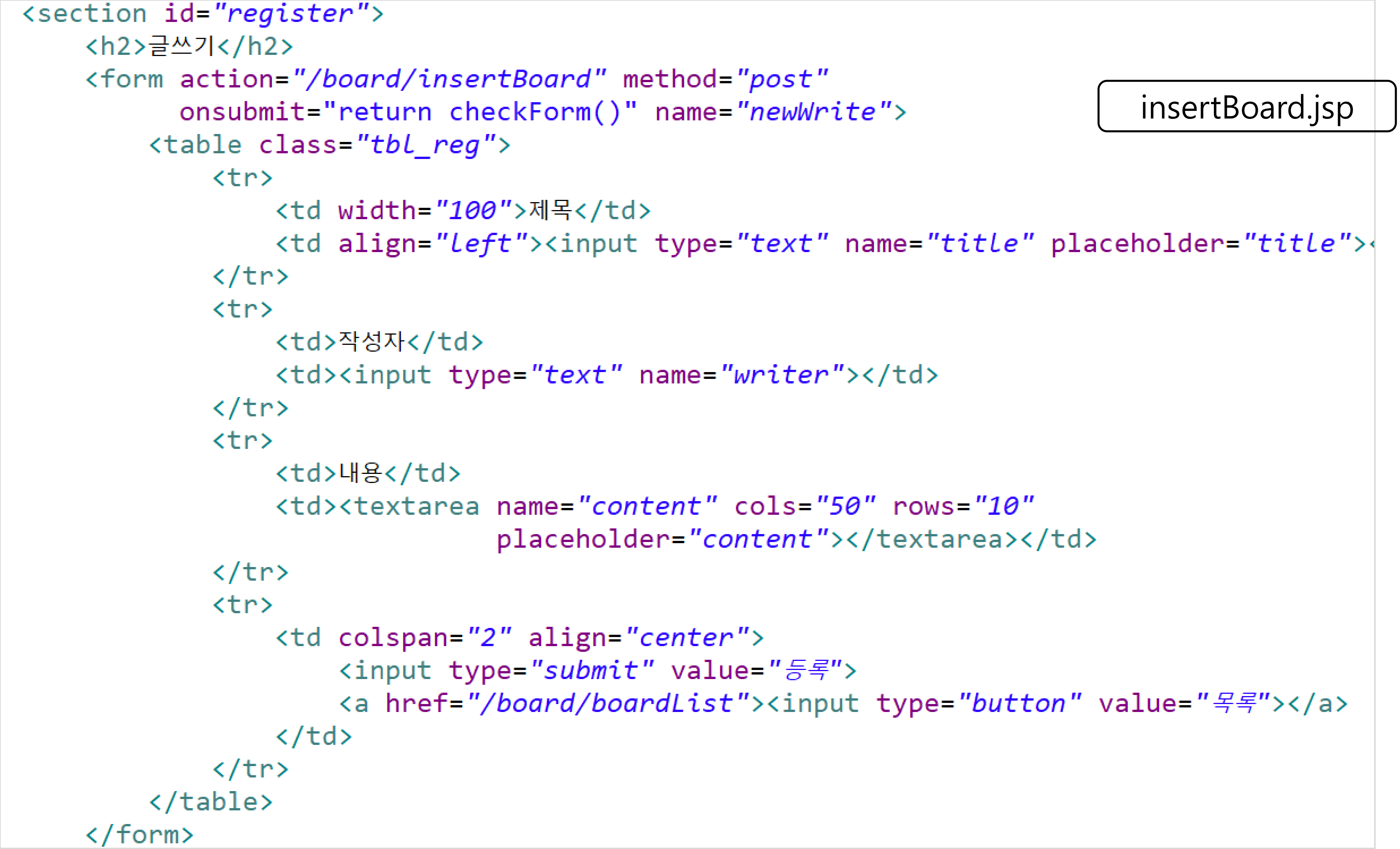


글쓰기

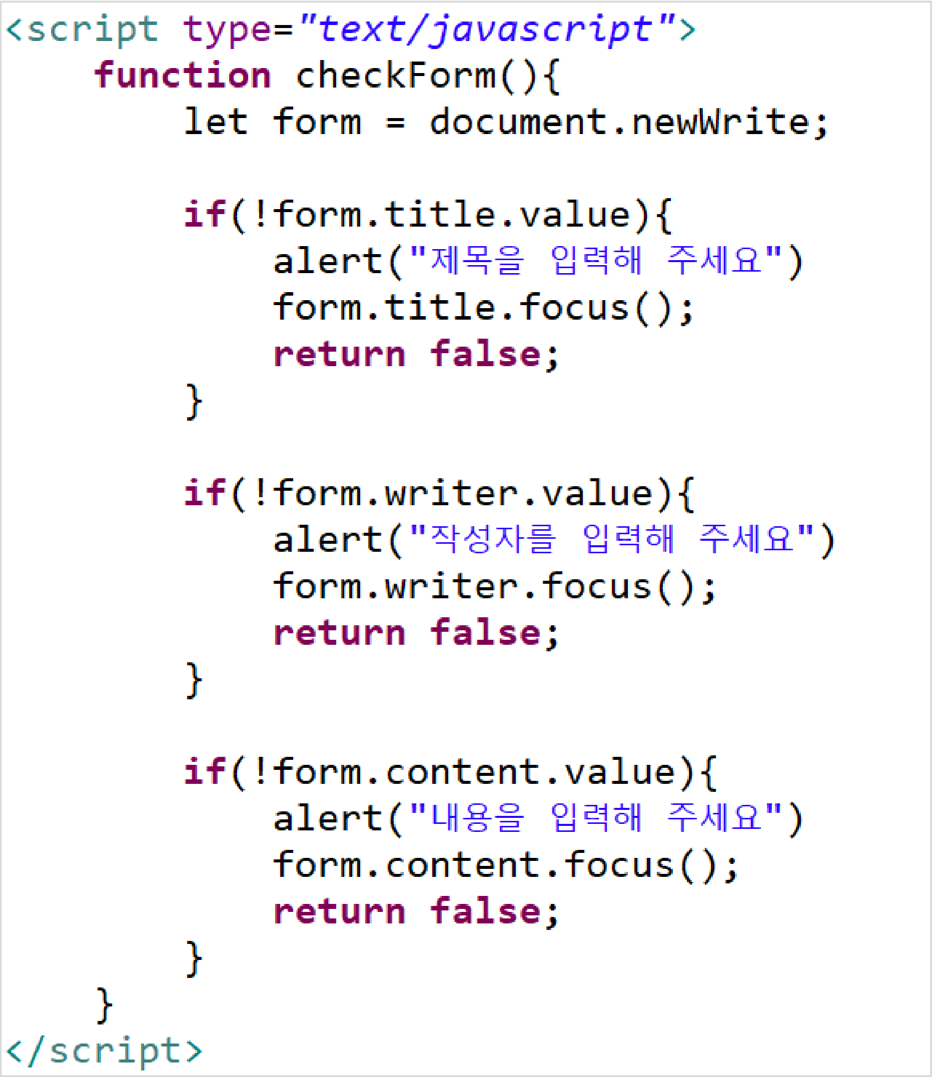
제목	title
작성자	
내용	content

SpringBoard

insertBoard.jsp

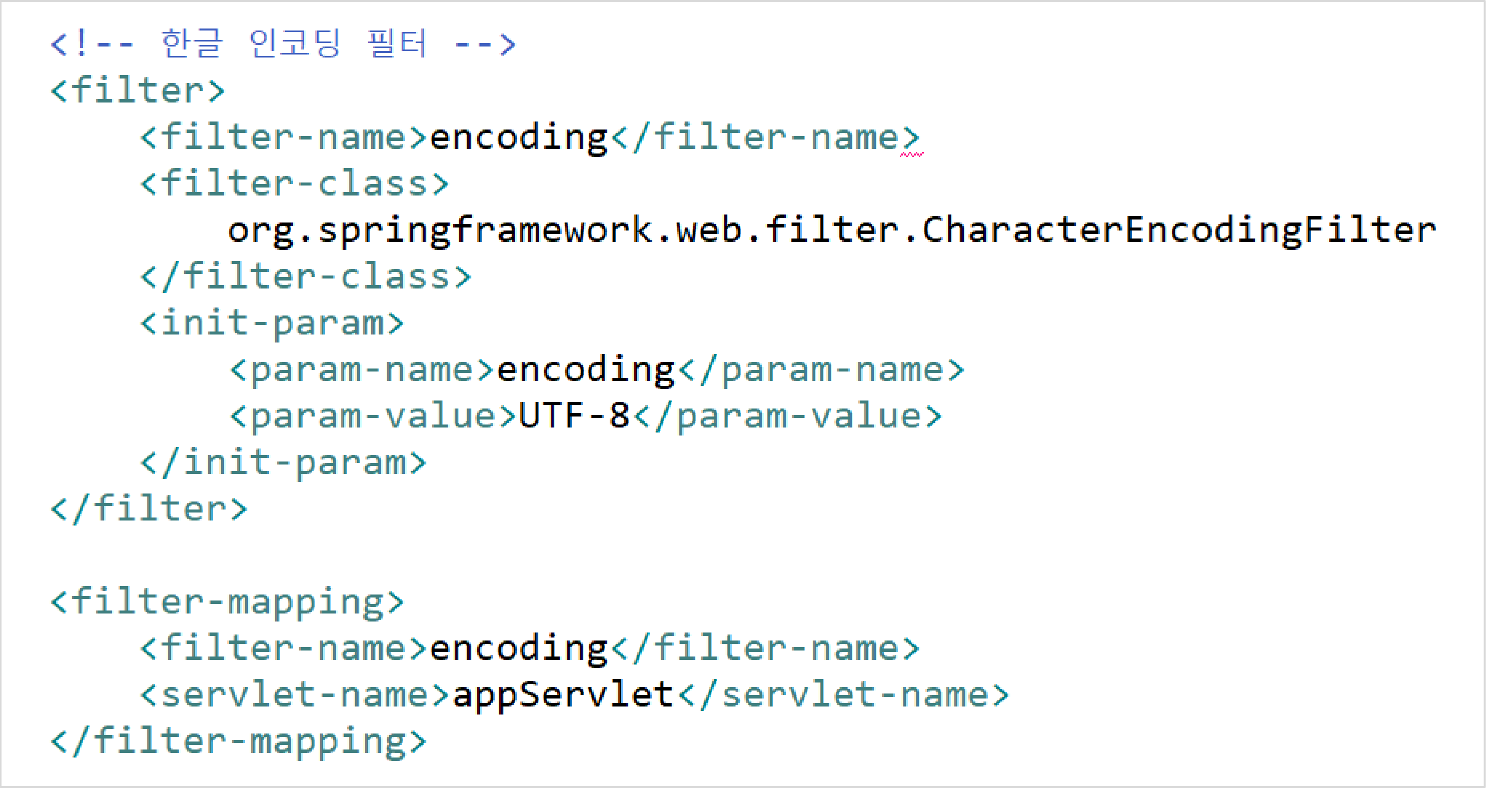


```
<section id="register">
    <h2>글쓰기</h2>
    <form action="/board/insertBoard" method="post"
          onsubmit="return checkForm()" name="newWrite">
        <table class="tbl_reg">
            <tr>
                <td width="100">제목</td>
                <td align="left"><input type="text" name="title" placeholder="title"><
            </tr>
            <tr>
                <td>작성자</td>
                <td><input type="text" name="writer"></td>
            </tr>
            <tr>
                <td>내용</td>
                <td><textarea name="content" cols="50" rows="10"
                              placeholder="content"></textarea></td>
            </tr>
            <tr>
                <td colspan="2" align="center">
                    <input type="submit" value="등록">
                    <a href="/board/boardList"><input type="button" value="목록"></a>
                </td>
            </tr>
        </table>
    </form>
```

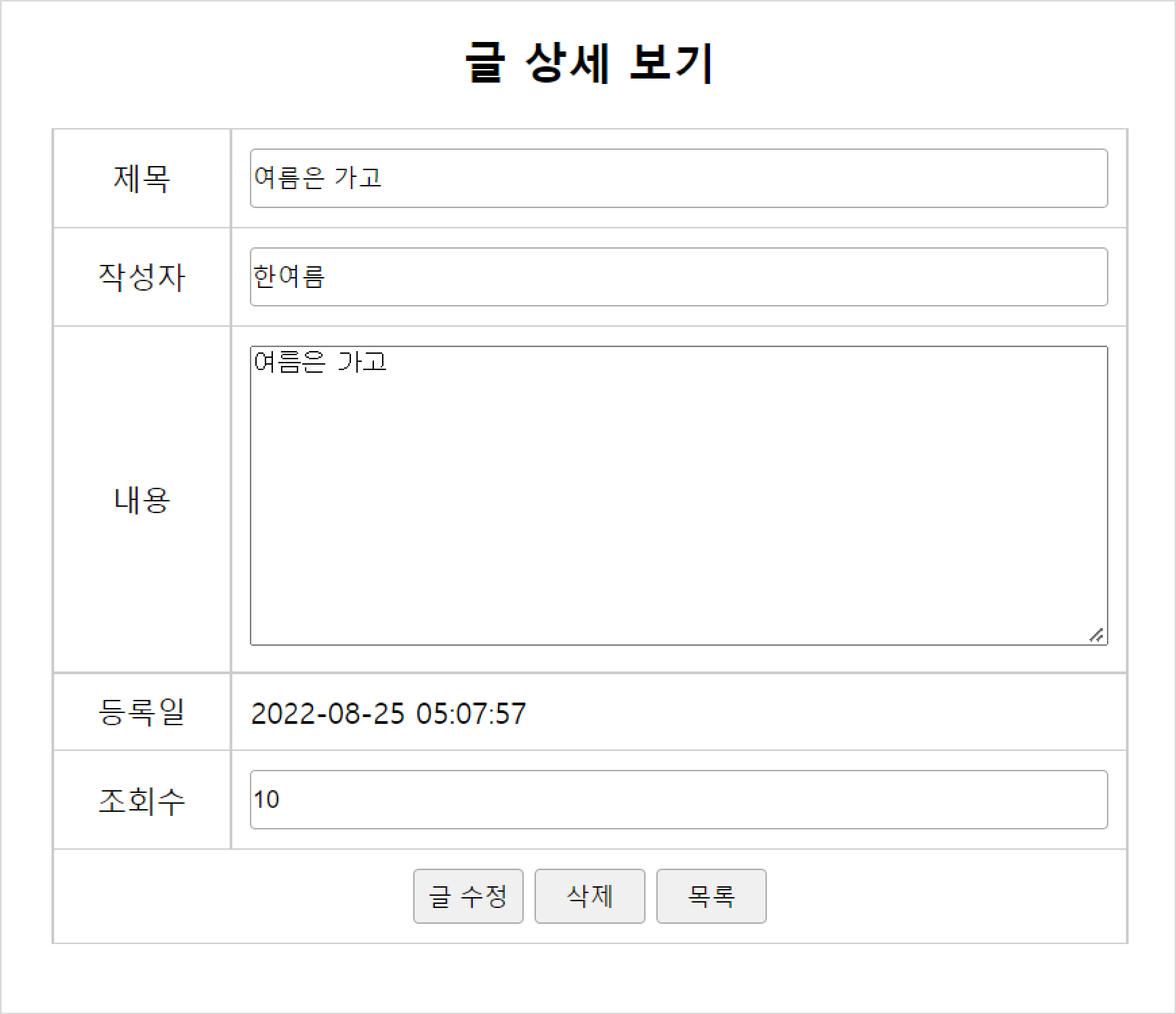


```
<script type="text/javascript">
    function checkForm(){
        let form = document.newWrite;

        if(!form.title.value){
            alert("제목을 입력해 주세요")
            form.title.focus();
            return false;
        }

        if(!form.writer.value){
            alert("작성자를 입력해 주세요")
            form.writer.focus();
            return false;
        }

        if(!form.content.value){
            alert("내용을 입력해 주세요")
            form.content.focus();
            return false;
        }
    }
</script>
```

- **한글 인코딩 처리(Filter) - web.xml**



```xml
<!-- 한글 인코딩 필터 -->
<filter>
    <filter-name>encoding</filter-name>
    <filter-class>
        org.springframework.web.filter.CharacterEncodingFilter
    </filter-class>
    <init-param>
        <param-name>encoding</param-name>
        <param-value>UTF-8</param-value>
    </init-param>
</filter>

<filter-mapping>
    <filter-name>encoding</filter-name>
    <servlet-name>appServlet</servlet-name>
</filter-mapping>
```

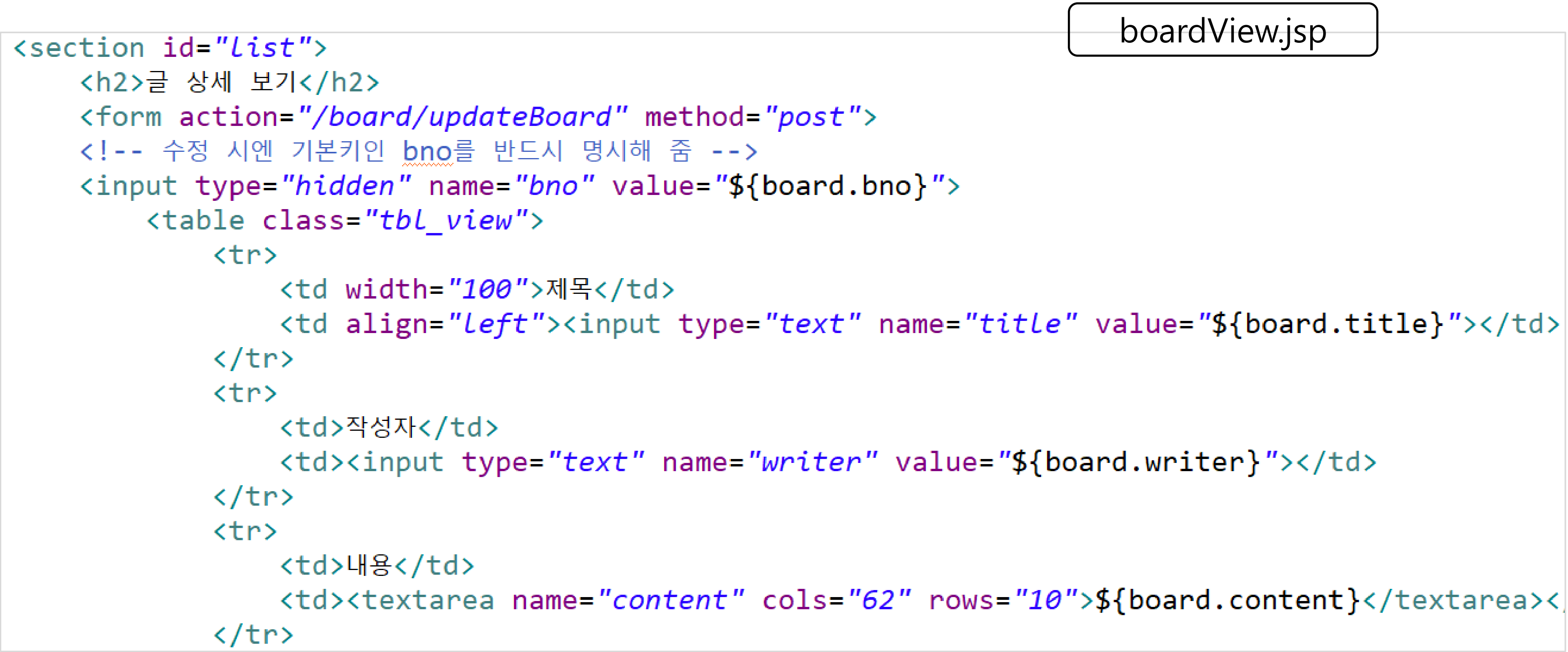


글 상세 보기

제목	여름은 가고
작성자	한여름
내용	여름은 가고
등록일	2022-08-25 05:07:57
조회수	10

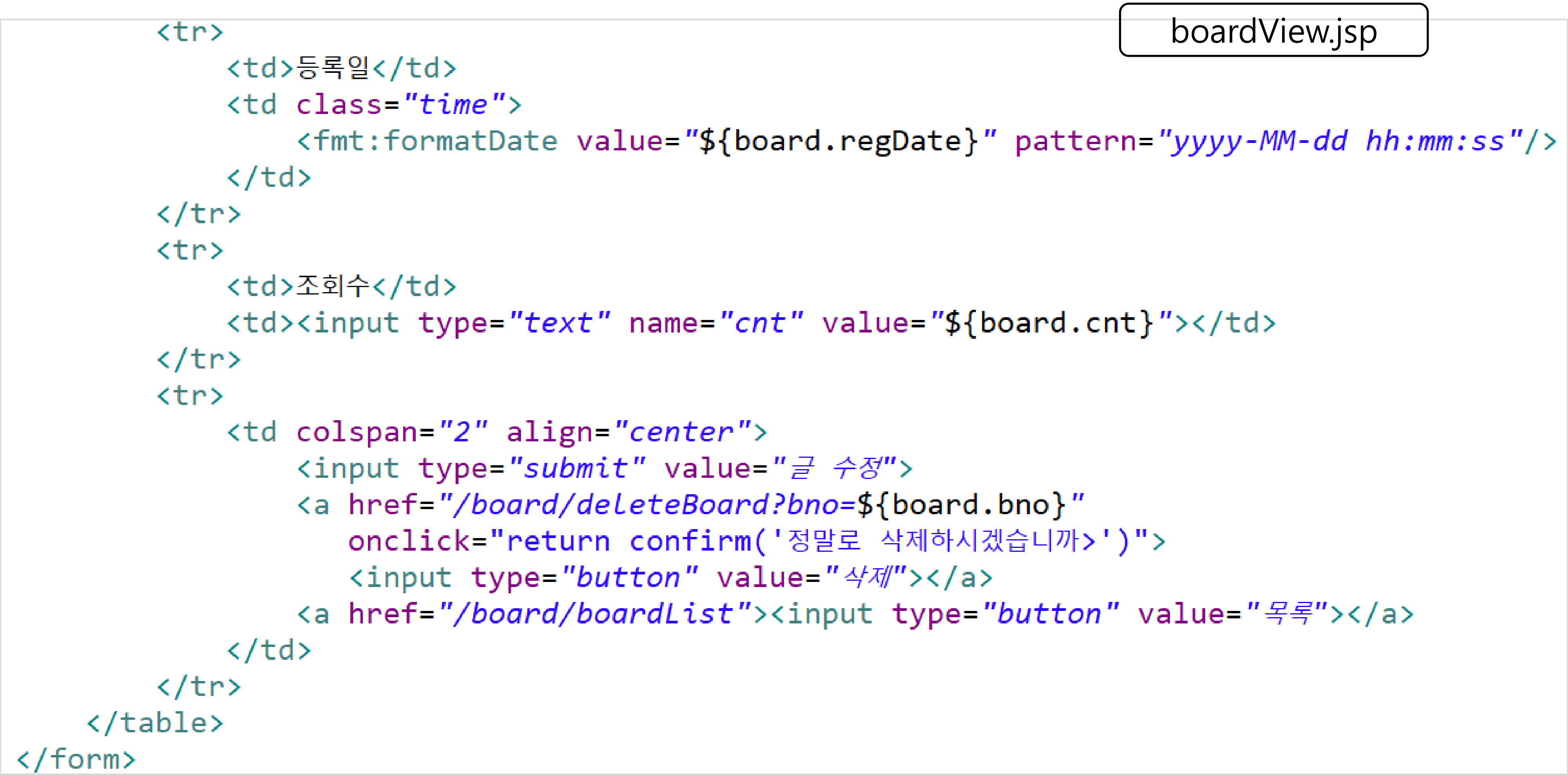
SpringBoard

boardView.jsp



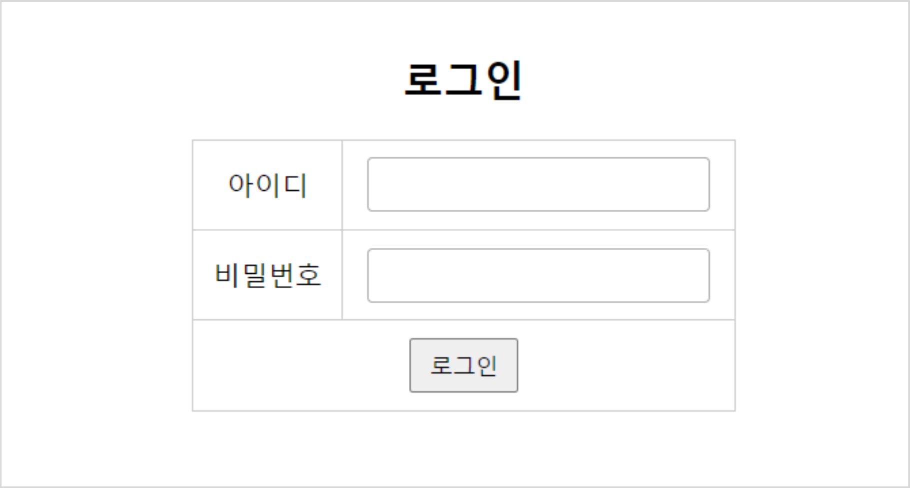
```
<section id="list">
    <h2>글 상세 보기</h2>
    <form action="/board/updateBoard" method="post">
    <!-- 수정 시엔 기본키인 bno를 반드시 명시해 줌 -->
    <input type="hidden" name="bno" value="${board.bno}">
        <table class="tbl_view">
            <tr>
                <td width="100">제목</td>
                <td align="left"><input type="text" name="title" value="${board.title}"></td>
            </tr>
            <tr>
                <td>작성자</td>
                <td><input type="text" name="writer" value="${board.writer}"></td>
            </tr>
            <tr>
                <td>내용</td>
                <td><textarea name="content" cols="62" rows="10">${board.content}</textarea><
            </tr>
```

SpringBoard



boardView.jsp

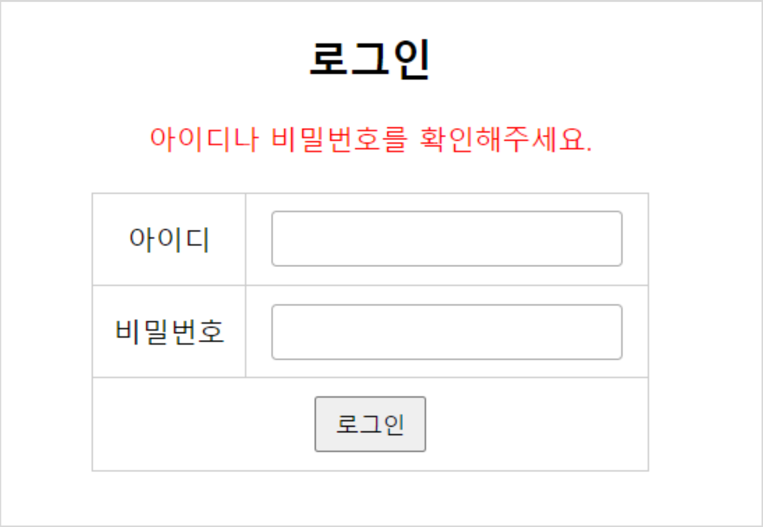
```
        <tr>
            <td>등록일</td>
            <td class="time">
                <fmt:formatDate value="${board.regDate}" pattern="yyyy-MM-dd hh:mm:ss"/>
            </td>
        </tr>
        <tr>
            <td>조회수</td>
            <td><input type="text" name="cnt" value="${board.cnt}"></td>
        </tr>
        <tr>
            <td colspan="2" align="center">
                <input type="submit" value="글 수정">
                <a href="/board/deleteBoard?bno=${board.bno}"
                   onclick="return confirm('정말로 삭제하시겠습니까>')">
                   <input type="button" value="삭제"></a>
                <a href="/board/boardList"><input type="button" value="목록"></a>
            </td>
        </tr>
    </table>
</form>
```



로그인

아이디	
비밀번호	

로그인



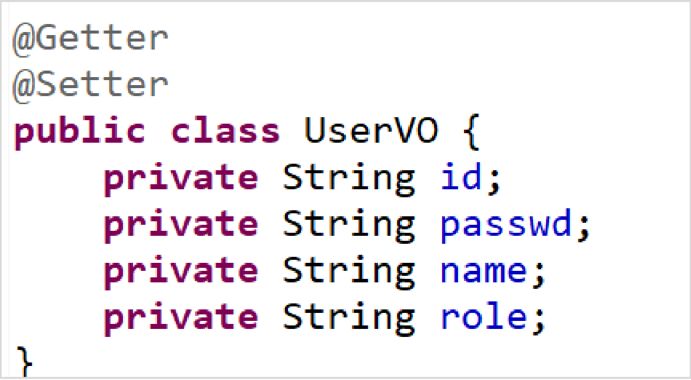
로그인

아이디나 비밀번호를 확인해주세요.

아이디	
비밀번호	

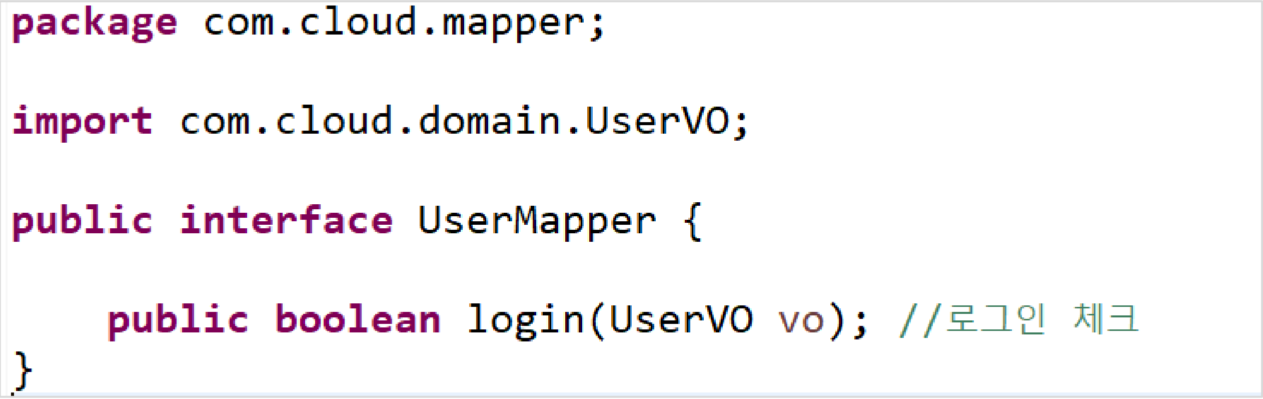
로그인

- VO



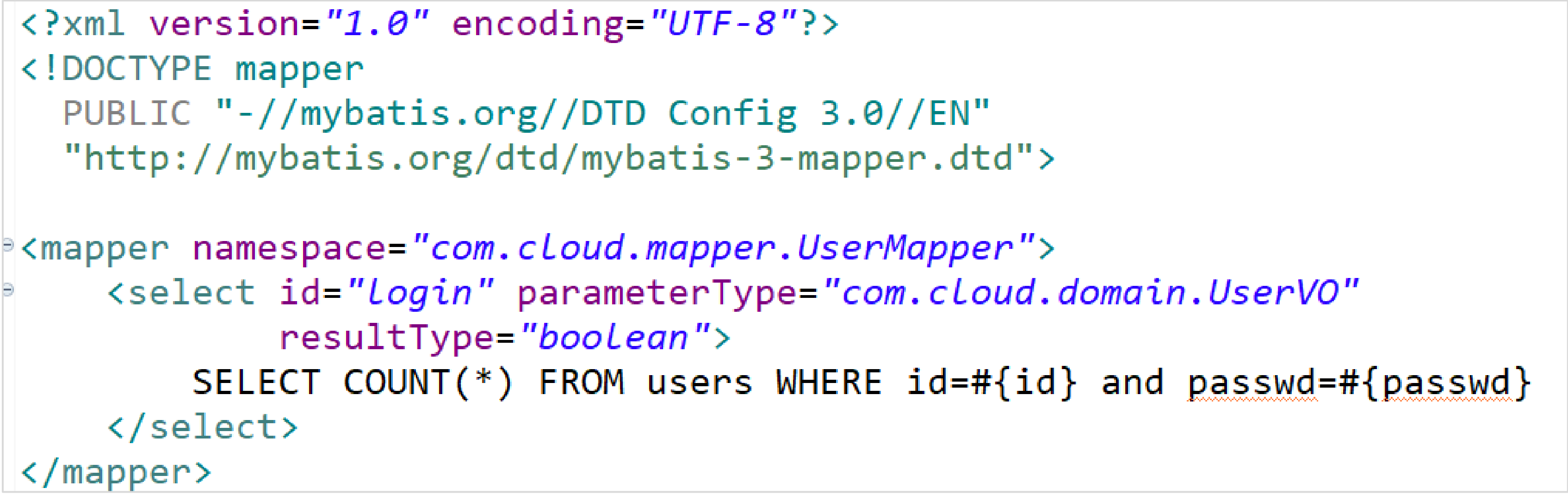
```
@Getter
@Setter
public class UserVO {
    private String id;
    private String passwd;
    private String name;
    private String role;
}
```

- **Mapper 인터페이스 생성 – Mapper.java**



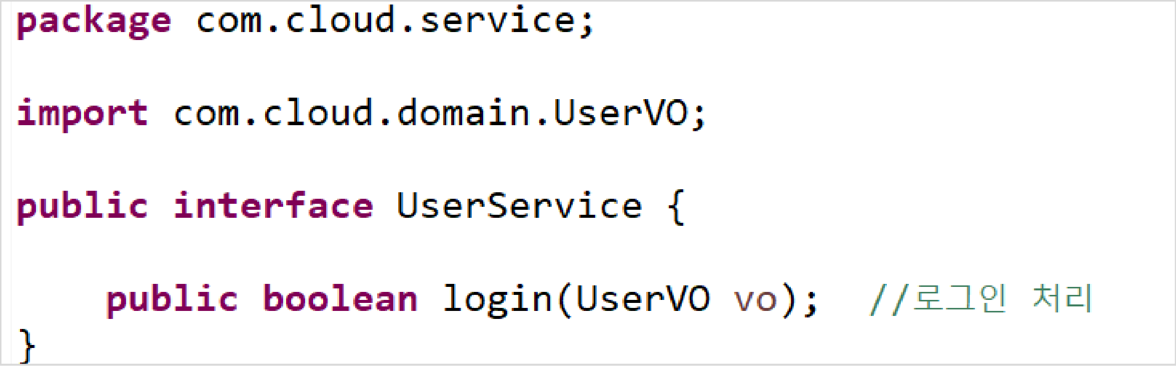
```java
package com.cloud.mapper;

import com.cloud.domain.UserVO;

public interface UserMapper {

    public boolean login(UserVO vo); //로그인 체크
}
```

- **Mapper 생성 – Mapper.xml**



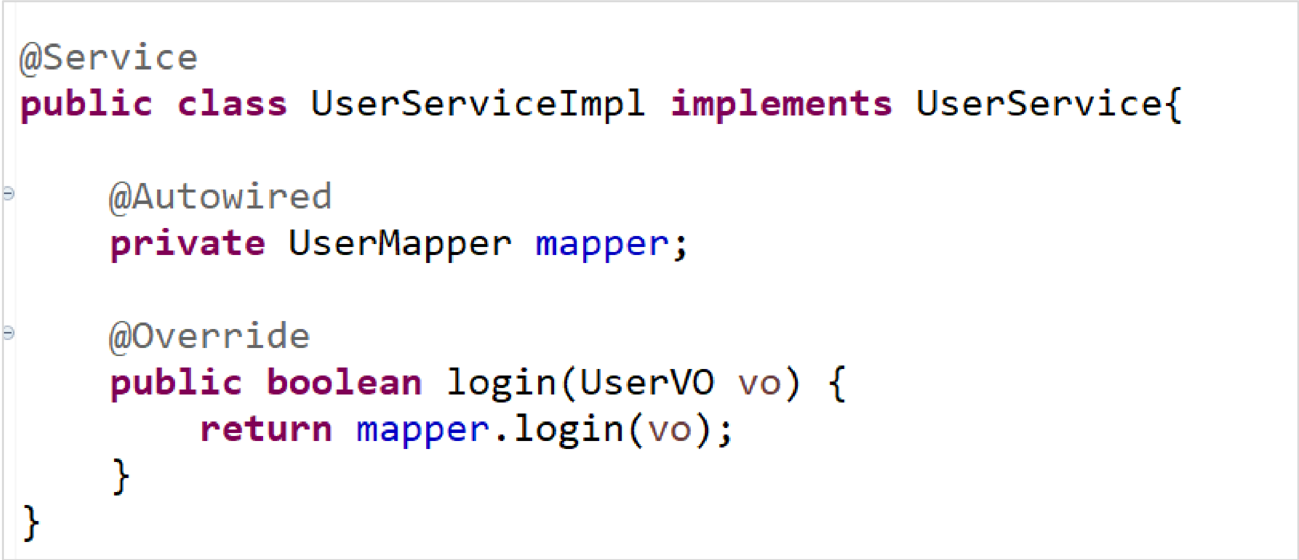
```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.cloud.mapper.UserMapper">
    <select id="login" parameterType="com.cloud.domain.UserVO"
            resultType="boolean">
        SELECT COUNT(*) FROM users WHERE id=#{id} and passwd=#{passwd}
    </select>
</mapper>
```

- **Service 계층**



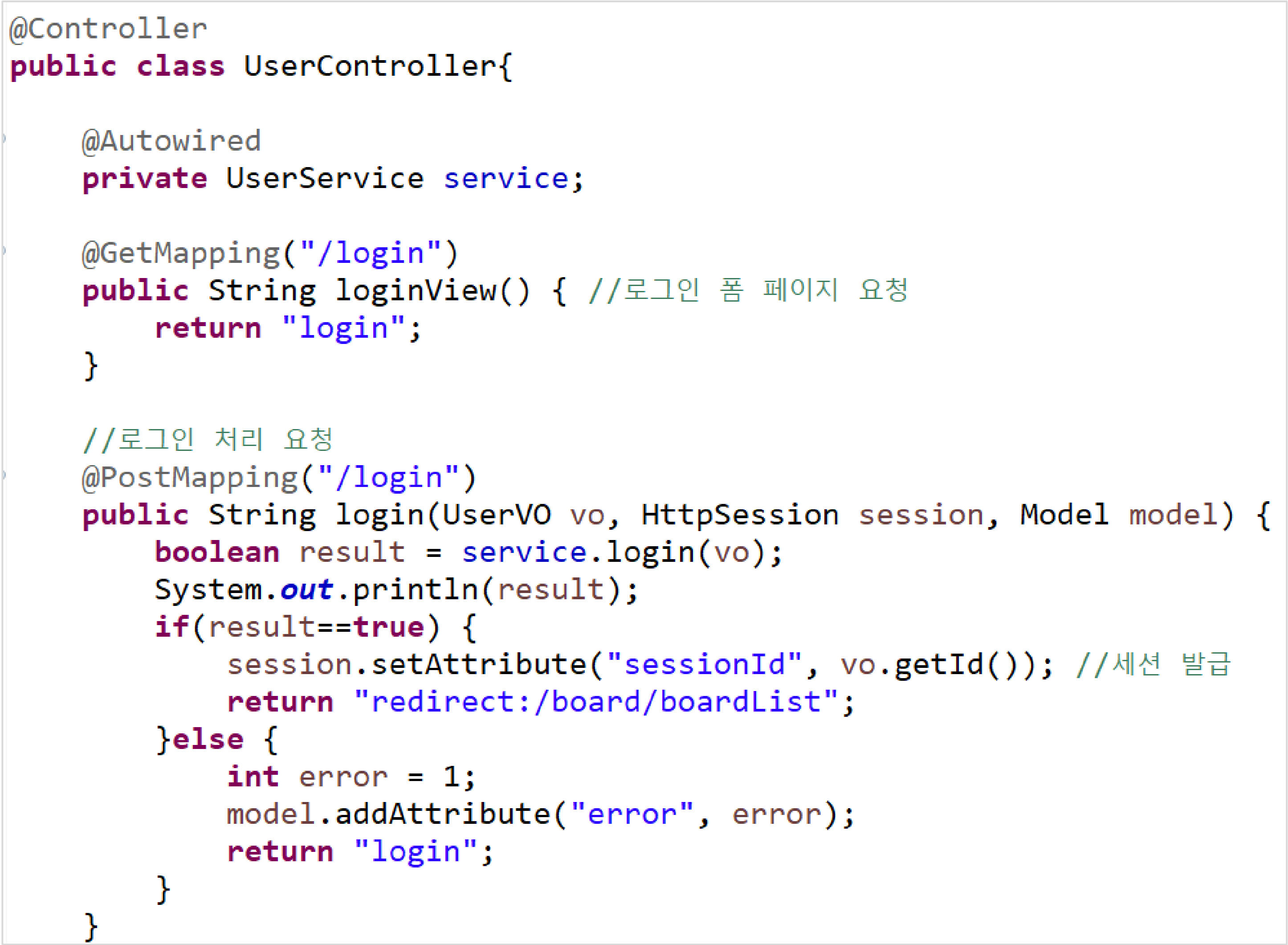
```java
package com.cloud.service;

import com.cloud.domain.UserVO;

public interface UserService {

    public boolean login(UserVO vo);  //로그인 처리
}
```

- **Service 계층 – UserServiceImpl.java**



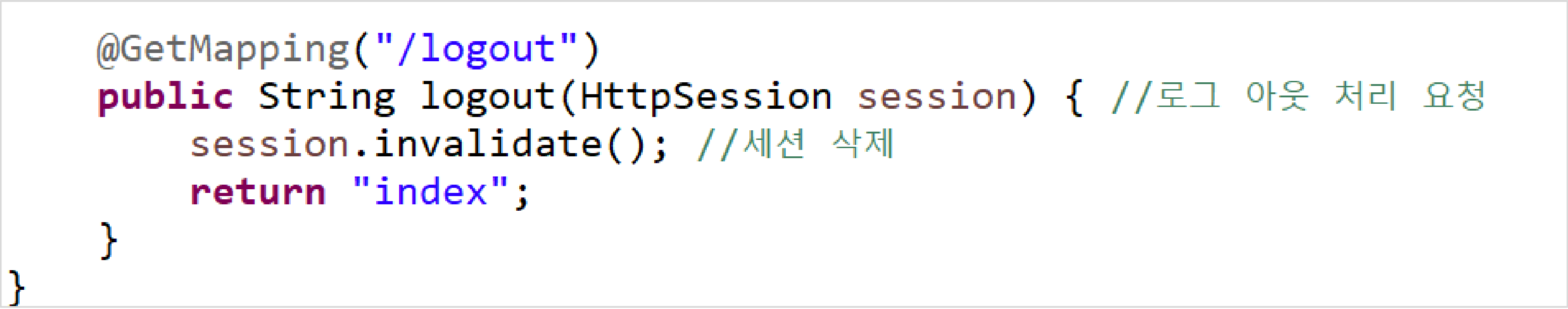
```java
@Service
public class UserServiceImpl implements UserService{

    @Autowired
    private UserMapper mapper;

    @Override
    public boolean login(UserVO vo) {
        return mapper.login(vo);
    }
}
```

- **Controller 계층 - UserController.java**

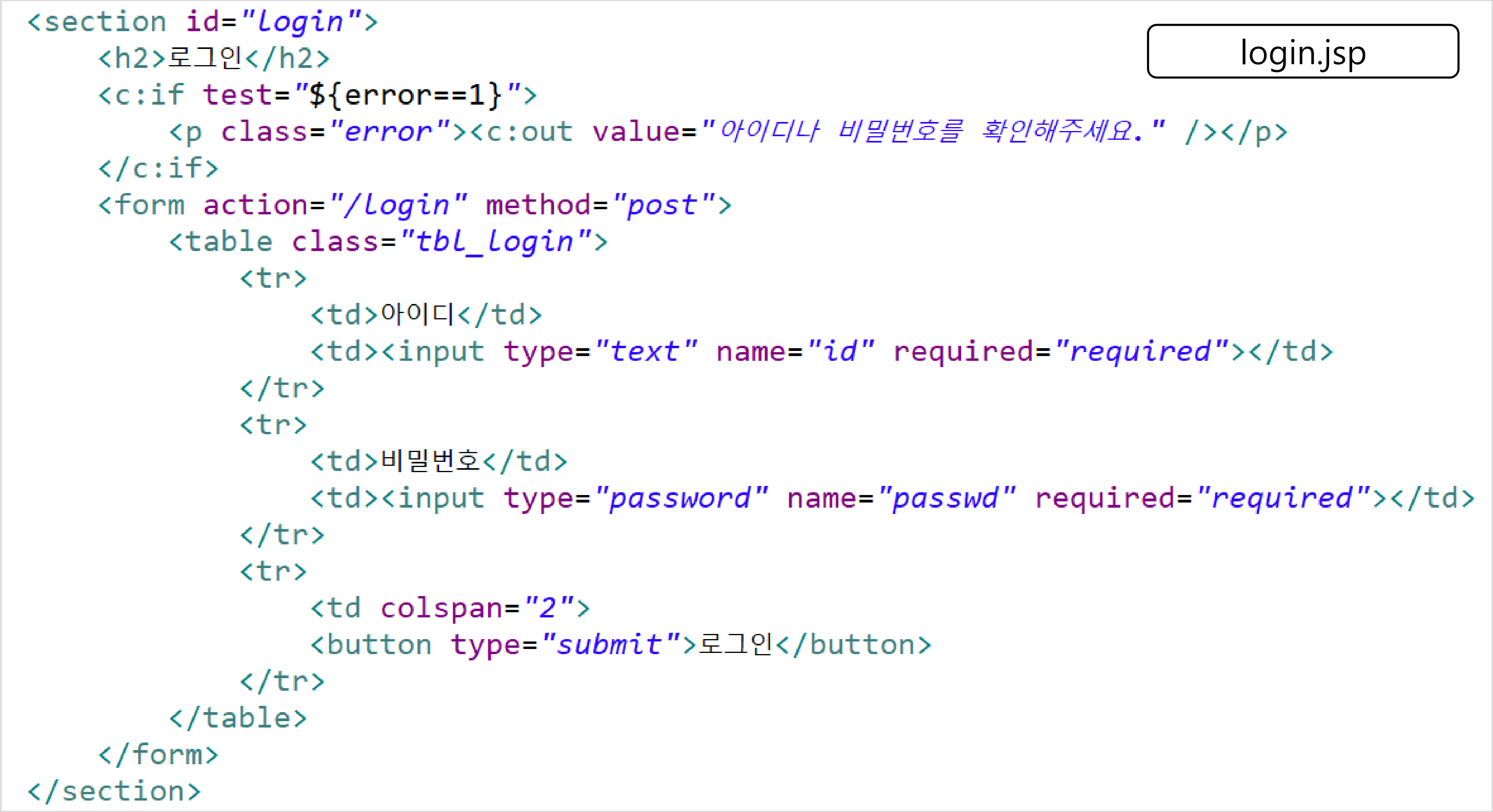


```java
@Controller
public class UserController{

    @Autowired
    private UserService service;

    @GetMapping("/login")
    public String loginView() { //로그인 폼 페이지 요청
        return "login";
    }

    //로그인 처리 요청
    @PostMapping("/login")
    public String login(UserVO vo, HttpSession session, Model model) {
        boolean result = service.login(vo);
        System.out.println(result);
        if(result==true) {
            session.setAttribute("sessionId", vo.getId()); //세션 발급
            return "redirect:/board/boardList";
        }else {
            int error = 1;
            model.addAttribute("error", error);
            return "login";
        }
    }
```

- **Controller 계층**



```
    @GetMapping("/logout")
    public String logout(HttpSession session) { //로그 아웃 처리 요청
        session.invalidate(); //세션 삭제
        return "index";
    }
}
```



login.jsp

```
<section id="login">
    <h2>로그인</h2>
    <c:if test="${error==1}">
        <p class="error"><c:out value="아이디나 비밀번호를 확인해주세요." /></p>
    </c:if>
    <form action="/login" method="post">
        <table class="tbl_login">
            <tr>
                <td>아이디</td>
                <td><input type="text" name="id" required="required"></td>
            </tr>
            <tr>
                <td>비밀번호</td>
                <td><input type="password" name="passwd" required="required"></td>
            </tr>
            <tr>
                <td colspan="2">
                <button type="submit">로그인</button>
            </tr>
        </table>
    </form>
</section>
```