

김운영

(디지털컨버전스) 스마트 콘텐츠와 웹 융합 응용SW개발자 양성
과정

[강남 C] 2021. 05. 07 ~ 2021. 12. 09 15:30~22:00

이상훈 강사 | 이은정 취업담임

남은 시간 01:40:50

수강생 평가

* 첨부파일의 확장자를 소문자로 등록하세요. - 예시 : test.jpg(O), test.JPG(X)

다음과 같은 요구사항에 맞춰 구성하여 보고서 작성 후 제출하시오.

서버프로그램에서 사용자의 게시판 글 등록 서비스를 구현하시오.

(처리결과 성공시에는 게시물 목록을 출력하는 blist 서블릿을 실행하고, 실패시에는 content directory/views/common/error.jsp 를 view 페이지로 지정하고 "message" 키로 "게시글 등록 실패!" 메시지를 포워딩 하시오.

데이터베이스의 board테이블 생성 스크립트는 다음과 같다.

```
CREATE TABLE BOARD(
BOARD_NUM NUMBER,
BOARD_WRITER VARCHAR2(20) NOT NULL,
BOARD_TITLE VARCHAR2(50) NOT NULL,
BOARD_CONTENT VARCHAR2(2000) NOT NULL,
BOARD_DATE DATE DEFAULT SYSDATE,
CONSTRAINT PK_BOARD PRIMARY KEY (BOARD_NUM),
CONSTRAINT FK_BOARD_WRITER FOREIGN KEY (BOARD_WRITER) REFERENCES MEMBER (USERID) ON DELETE SET NULL
);
COMMENT ON COLUMN BOARD.BOARD_NUM IS '게시글번호';
COMMENT ON COLUMN BOARD.BOARD_WRITER IS '작성자아이디';
COMMENT ON COLUMN BOARD.BOARD_TITLE IS '게시글제목';
COMMENT ON COLUMN BOARD.BOARD_CONTENT IS '게시글내용';
COMMENT ON COLUMN BOARD.BOARD_DATE IS '작성날짜';
사용자에게 제공할 게시판 등록 폼페이지는 다음과 같다.
<form action="<%= request.getContextPath() %>/binsert" method="post">
<table align="center">
<tr><td>제목</td>
<td><input type="text" name="btitle"></td></tr>
<tr><td>작성자</td>
<td><input type="text" readonly name="bwriter"
value="<%= loginUser.getUserId() %>"></td></tr>
<tr><td>내용</td>
<td><textarea cols="50" rows="7" name="bcontent"></textarea></td></tr>
<tr><td colspan="2" align="center">
<input type="submit" value="등록하기">
<a href="/first/blist?page=1">목록으로</a></td></tr>
</table>
</form>
```

게시글 등록 서블릿 클래스의 doPost()와 doGet() 메소드 안의 코드를 작성하시오.

BoardWriteServlet.java 의 doPost()와 doGet() 을 구현하시오. [EDIT](#)

BoardDao.java 클래스의 insertBoard(conn, board) 메소드를 PreparedStatement 로 새 게시물 등록 처리가 되도록 구현하시오.

BoardDao.java 의 insertBoard() 메소드를 구현하시오. [EDIT](#)

```
public void insertBoard(BoardVO vo) {
    System.out.println;
    try {
        conn = JDBCUtil.getConnection();
        stmt = conn.prepareStatement(BOARD_INSERT);
        stmt.setString(1, vo.getTitle());
        stmt.setString(2, vo.getWriter());
        stmt.setString(3, vo.getContent());
        stmt.executeUpdate();
    } catch (Exception e) {
        e.printStackTrace();
    } finally {
        JDBCUtil.close(stmt, conn);
    }
}
```

공통도틀인 JDBCTemplate.java 의 getConnect() 을 사용하여 Connection 을 생성하고, BoardDao 클래스의 insertBoard() 로 Connection 과 전달받은 Board 객체 정보를 전달하고 나서 처리결과를 리턴받아 트랜잭션을 처리하고 Connection 을 close 하고 받은 결과를 컨트롤러로 리턴한다.

공통모듈로 JDBCTemplate.java 를 싱글톤 패턴을 사용하여 구현하시오.

- 오라클과 연동 처리한다. (IP: 128.168.25.30, SID: XE)
- Connection 을 리턴하는 getConnection() 메소드를 구현한다.
- close() 메소드를 구현한다. 각각 Connection, Statement, ResultSet 에 대해 overloading 처리한다.
- commit, rollback 메소드도 구현한다.

JDBCTemplate.java 클래스를 구현하시오.

[EDIT](#)

```
public class JDBCTemplate {

    public static Connection getConnection() {
        try {
            Class.forName("oracle.jdbc.driver.OracleDriver");
        } catch (ClassNotFoundException e) {
            System.out.println("[Error] : JDBCTemplate.Driver");
            e.printStackTrace();
        }

        String url = "jdbc:oracle:thin:@localhost:128.168.25.30:xe";
        String user = "kh";
        String password = "kh";

        Connection con = null;
        try {

            con = DriverManager.getConnection(url, user, password);
            con.setAutoCommit(false);

        } catch (SQLException e) {
            System.out.println("[Error] : JDBCTemplate.getConnection");
            e.printStackTrace();
        }

        return con;
    }

    public static void close(ResultSet rs) {
        try {
            rs.close();
        } catch (SQLException e) {
            System.out.println("[Error] : JDBCTemplate.close(rs)");
            e.printStackTrace();
        }
    }

    public static void close(Statement stmt) {
        try {
            stmt.close();
        } catch (SQLException e) {
            System.out.println("[Error] : JDBCTemplate.close(stmt)");
            e.printStackTrace();
        }
    }

    public static void close(Connection con) {
        try {
            con.close();
        } catch (SQLException e) {
            System.out.println("[Error] : JDBCTemplate.close(con)");
            e.printStackTrace();
        }
    }

    public static void commit(Connection con) {
        try {
            con.commit();
        } catch (SQLException e) {
            System.out.println("[Error] : JDBCTemplate.commit(con)");
            e.printStackTrace();
        }
    }

    public static void rollback(Connection con) {
        try {
            con.rollback();
        } catch (SQLException e) {
            System.out.println("[Error] : JDBCTemplate.rollback(con)");
            e.printStackTrace();
        }
    }

}
```

