

통합 구현

2022-11-18 정기 수행평가

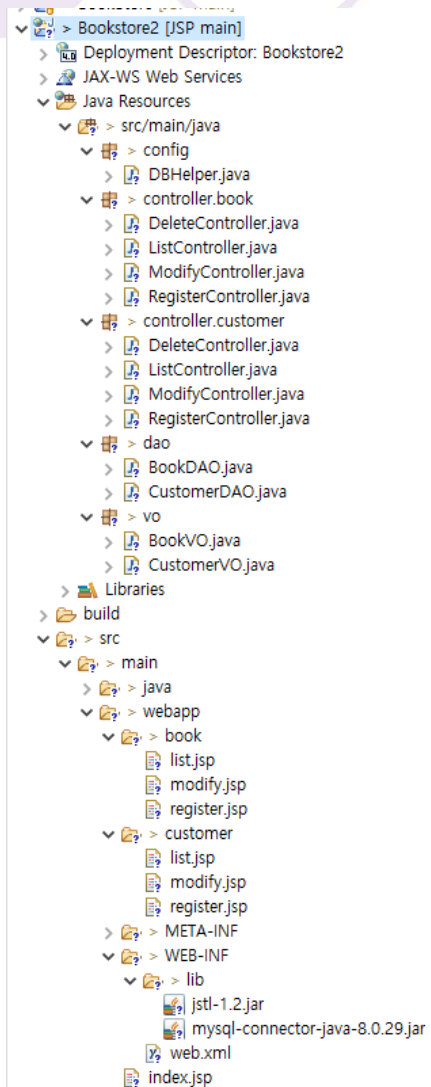
홍민준

목차

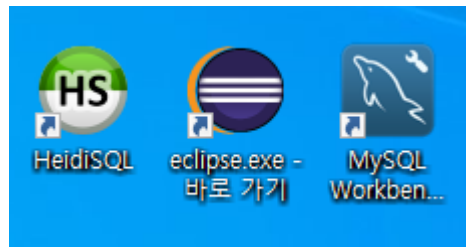
- ⦿ 문제 1번. 프로젝트 생성 및 구성
- ⦿ 문제 2번. 화면 구현
- ⦿ 문제 3번. 기능 구현
- ⦿ 문제 4번. 실행

Q 1 - 프로젝트 생성 및 구성

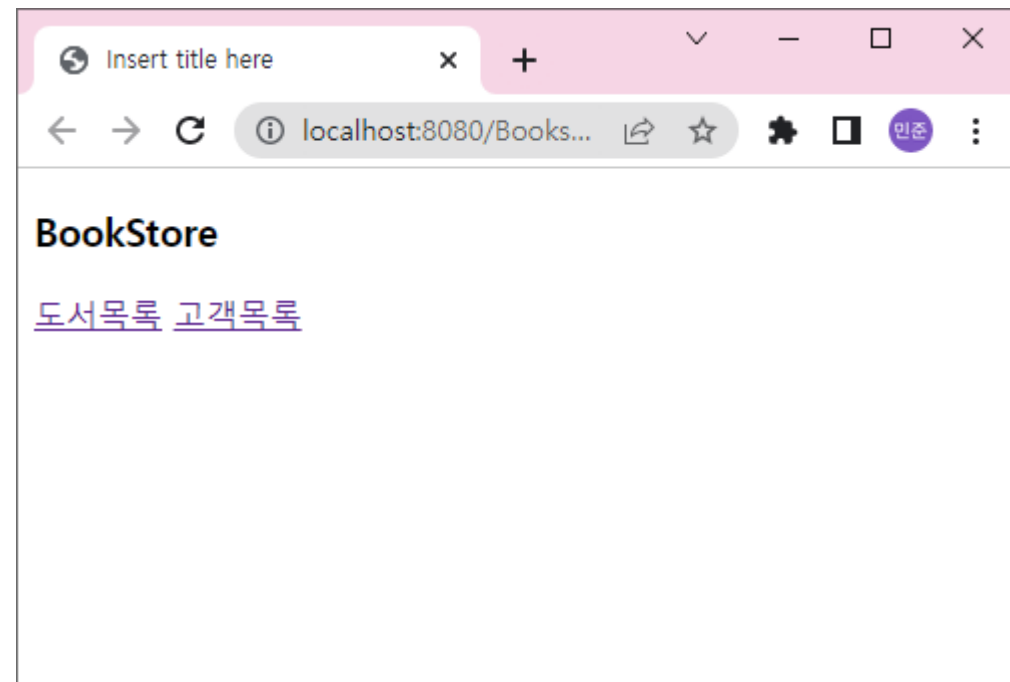
프로젝트 구조



개발도구



시작페이지



Q 2 - book 화면 구현

⦿ book 목록

⦿ book 등록

⦿ book 수정

Book::list

localhost:8080/Books...

도서 목록

[처음으로 도서등록](#)

도서번호	도서명	출판사	가격	관리
2	축구의 이해	대한미디어	11001	수정 삭제
3	골프바이블	대한미디어	35000	수정 삭제

book::register

localhost:8080/Books...

도서등록

[처음으로 도서목록](#)

도서번호	도서번호 입력
도서명	도서명 입력
출판사	출판사 입력
가격	가격 입력
<input type="button" value="등록"/>	

book::modify

localhost:8080/Books...

도서수정

[도서목록](#)

도서번호	2
도서명	축구의 이해
출판사	대한미디어
가격	11001
<input type="button" value="수정"/>	

Q 2 - Customer 화면 구현

Customer 목록

Customer::list

localhost:8080/Boo...

고객 목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
3	장미	대한민국 강원도맨체스터	010-1234-1003	수정 삭제
5	장발장	영국	010-1234-1234	수정 삭제

Customer 등록

customer::register

localhost:8080/Boo...

고객등록

처음으로 고객목록

고객명	<input type="text" value="고객명 입력"/>
주소	<input type="text" value="주소 입력"/>
휴대폰	<input type="text" value="휴대폰 입력"/>
<input type="button" value="등록"/>	

Customer 수정

customer::modify

localhost:8080/Boo...

고객수정

고객목록

고객번호	<input type="text" value="3"/>
고객명	<input type="text" value="장미"/>
주소	<input type="text" value="대한민국 강원도맨체스터"/>
휴대폰	<input type="text" value="010-1234-1003"/>
<input type="button" value="수정"/>	

Q 3 - book 기능 구현

◉ Lecture 코드:list

```
package controller.book;

import java.io.IOException;

@WebServlet("/book/list.do")
public class ListController extends HttpServlet{

    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {
    }
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        List<BookVO> books = BookDAO.getInstance().selectBooks();

        System.out.println("books : "+books.size());

        // view에서 데이터 출력을 위한 request Scope 데이터 설정
        req.setAttribute("books", books);

        // forward
        RequestDispatcher dispatcher = req.getRequestDispatcher("/book/list.jsp");
        dispatcher.forward(req, resp);

    }
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    }
}
```

◉ Lecture 코드:delete

```
package controller.book;

import java.io.IOException;

@WebServlet("/book/delete.do")
public class DeleteController extends HttpServlet{

    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {
    }

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String bno = req.getParameter("bno");
        BookDAO.getInstance().deleteBook(Integer.parseInt(bno));

        // 리다이렉트
        resp.sendRedirect("/Bookstore2/book/list.do");
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    }
}
```

◉ Lecture 코드:register

```
package controller.book;

import java.io.IOException;

@WebServlet("/book/register.do")
public class RegisterController extends HttpServlet{

    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {
    }
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        // forward
        RequestDispatcher dispatcher = req.getRequestDispatcher("/book/register.jsp");
        dispatcher.forward(req, resp);
    }
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        // POST 요청
        String bno = req.getParameter("bno");
        String bname = req.getParameter("bname");
        String bpub = req.getParameter("bpub");
        String bpric = req.getParameter("bpric");

        System.out.println("파라미터 get");

        BookVO vo = new BookVO();
        vo.setBno(bno);
        vo.setBname(bname);
        vo.setBpub(bpub);
        vo.setBpric(bpric);

        BookDAO.getInstance().insertBook(vo);

        //리다이렉트
        resp.sendRedirect("/Bookstore2/book/list.do");
    }
}
```

◉ Lecture 코드:modify

```
package controller.book;

import java.io.IOException;

@WebServlet("/book/modify.do")
public class ModifyController extends HttpServlet{

    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {
    }
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        String bno = req.getParameter("bno");
        BookVO vo = BookDAO.getInstance().selectBook(Integer.parseInt(bno));

        req.setAttribute("vo", vo);

        // 포워드
        RequestDispatcher dispatcher = req.getRequestDispatcher("/book/modify.jsp");
        dispatcher.forward(req, resp);
    }
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String bno = req.getParameter("bno");
        String bname = req.getParameter("bname");
        String bpub = req.getParameter("bpub");
        String bpric = req.getParameter("bpric");

        BookVO vo = new BookVO();
        vo.setBno(bno);
        vo.setBname(bname);
        vo.setBpub(bpub);
        vo.setBpric(bpric);

        BookDAO.getInstance().updateBook(vo);

        // 리다이렉트
        resp.sendRedirect("/Bookstore2/book/list.do");
    }
}
```

Q 3 - customer 기능 구현

Customer코드:list

```
package controller.customer;

import java.io.IOException;

@WebServlet("/customer/list.do")
public class ListController extends HttpServlet{

    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {
    }
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        List<CustomerVO> customers = CustomerDAO.getInstance().selectCustomers();

        System.out.println("customers : "+customers.size());

        // view에서 데이터 출력을 위한 request Scope 데이터 설정
        req.setAttribute("customers", customers);

        // forward
        RequestDispatcher dispatcher = req.getRequestDispatcher("/customer/list.jsp");
        dispatcher.forward(req, resp);

    }
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    }
}
```

Customer코드:delete

```
package controller.customer;

import java.io.IOException;

@WebServlet("/customer/delete.do")
public class DeleteController extends HttpServlet{

    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {
    }

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String custId = req.getParameter("custId");
        CustomerDAO.getInstance().deleteCustomer(Integer.parseInt(custId));

        // 리다이렉트
        resp.sendRedirect("/Bookstore2/customer/list.do");
    }

    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    }
}
```

Customer코드:modify

```
package controller.customer;

import java.io.IOException;

@WebServlet("/customer/modify.do")
public class ModifyController extends HttpServlet{

    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {
    }
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {

        String custId = req.getParameter("custId");
        CustomerVO vo = CustomerDAO.getInstance().selectCustomer(Integer.parseInt(custId));

        req.setAttribute("vo", vo);

        // 포워드
        RequestDispatcher dispatcher = req.getRequestDispatcher("/customer/modify.jsp");
        dispatcher.forward(req, resp);
    }
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        String custId = req.getParameter("custId");
        String cusname = req.getParameter("cusname");
        String cusaddr = req.getParameter("cusaddr");
        String cushp = req.getParameter("cushp");

        CustomerVO vo = new CustomerVO();
        vo.setCustId(custId);
        vo.setCusname(cusname);
        vo.setCusaddr(cusaddr);
        vo.setCushp(cushp);

        CustomerDAO.getInstance().updateCustomer(vo);

        // 리다이렉트
        resp.sendRedirect("/Bookstore2/customer/list.do");
    }
}
```

Customer코드:register

```
package controller.customer;

import java.io.IOException;

@WebServlet("/customer/register.do")
public class RegisterController extends HttpServlet{

    private static final long serialVersionUID = 1L;

    @Override
    public void init() throws ServletException {
    }
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        // forward
        RequestDispatcher dispatcher = req.getRequestDispatcher("/customer/register.jsp");
        dispatcher.forward(req, resp);
    }
    @Override
    protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
        // POST 요청
        String cusname = req.getParameter("cusname");
        String cusaddr = req.getParameter("cusaddr");
        String cushp = req.getParameter("cushp");

        CustomerVO vo = new CustomerVO();
        vo.setCusname(cusname);
        vo.setCusaddr(cusaddr);
        vo.setCushp(cushp);

        CustomerDAO.getInstance().insertCustomer(vo);

        //리다이렉트
        resp.sendRedirect("/Bookstore2/customer/list.do");
    }
}
```

Q 3 - 기능 구현 DBHelper

```
DBHelper.java ×
1 package config;
2 import java.sql.Connection;
3
4
5
6
7
8
9 public class DBHelper {
10
11     protected Connection conn = null;
12     protected PreparedStatement psmt = null;
13     protected Statement stmt = null;
14     protected ResultSet rs = null;
15
16     public Connection getConnection() {
17         try {
18             DataSource ds = (DataSource) new InitialContext().lookup("java:comp/env/dbcp_java1_bookstore");
19             conn = ds.getConnection();
20         } catch (Exception e) {
21             e.printStackTrace();
22         }
23         return conn;
24     }
25
26     public void close() {
27         try {
28             if(rs != null) {
29                 rs.close();
30             }
31
32             if(stmt != null) {
33                 stmt.close();
34             }
35
36             if(psmt != null) {
37                 psmt.close();
38             }
39
40             if(conn != null) {
41                 conn.close();
42             }
43
44         } catch (Exception e) {
45             e.printStackTrace();
46         }
47     }
48 }
```


Q 3 - 기능 구현 DAO

CustomerDAO코드

```
package dao;
import java.sql.SQLException;

public class CustomerDAO extends DBHelper{
    private static CustomerDAO instance = new CustomerDAO();
    public static CustomerDAO getInstance() {
        return instance;
    }
    private CustomerDAO() {}

    public void insertCustomer(CustomerVO vo) {
        try {
            conn = getConnection();
            pstmt = conn.prepareStatement("insert into `customer` set `cusname`=?, `cusaddr`=?, `cushp`=?");
            pstmt.setString(1, vo.getCusname());
            pstmt.setString(2, vo.getCusaddr());
            pstmt.setString(3, vo.getCushp());
            pstmt.executeUpdate();
            close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public CustomerVO selectCustomer(int custId) {
        CustomerVO vo = null;
        try {
            conn = getConnection();
            pstmt = conn.prepareStatement("select * from `customer` where `custId`=?");
            pstmt.setInt(1, custId);
            rs = pstmt.executeQuery();

            if(rs.next()) {
                vo = new CustomerVO();
                vo.setCustId(rs.getInt(1));
                vo.setCusname(rs.getString(2));
                vo.setCusaddr(rs.getString(3));
                vo.setCushp(rs.getString(4));
            }
            close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return vo;
    }

    public List<CustomerVO> selectCustomers() {
        List<CustomerVO> customers = new ArrayList<>();
        try {
            conn = getConnection();
            stmt = conn.createStatement();
            rs = stmt.executeQuery("select * from `customer`");

            while(rs.next()) {
                CustomerVO vo = new CustomerVO();
                vo.setCustId(rs.getInt(1));
                vo.setCusname(rs.getString(2));
                vo.setCusaddr(rs.getString(3));
                vo.setCushp(rs.getString(4));
                customers.add(vo);
            }
            close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return customers;
    }

    public void updateCustomer(CustomerVO vo) {
        try {
            conn = getConnection();
            pstmt = conn.prepareStatement("update `customer` set `cusname`=?, `cusaddr`=?, `cushp`=? where `custId`=?");
            pstmt.setString(1, vo.getCusname());
            pstmt.setString(2, vo.getCusaddr());
            pstmt.setString(3, vo.getCushp());
            pstmt.setInt(4, vo.getCustId());
            pstmt.executeUpdate();
            close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public void deleteCustomer(int custId) {
        try {
            conn = getConnection();
            pstmt = conn.prepareStatement("delete from `customer` where `custId`=?");
            pstmt.setInt(1, custId);
            pstmt.executeUpdate();
            close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

BookDAO코드

```
package dao;
import java.sql.SQLException;

public class BookDAO extends DBHelper{
    private static BookDAO instance = new BookDAO();
    public static BookDAO getInstance() {
        return instance;
    }
    private BookDAO() {}

    public void insertBook(BookVO vo) {
        try {
            conn = getConnection();
            pstmt = conn.prepareStatement("insert into `book` values(?,?,?,?)");
            pstmt.setInt(1, vo.getBno());
            pstmt.setString(2, vo.getBname());
            pstmt.setString(3, vo.getBpub());
            pstmt.setInt(4, vo.getBpric());
            pstmt.executeUpdate();
            close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    public BookVO selectBook(int bno) {
        BookVO vo = null;
        try {
            conn = getConnection();
            pstmt = conn.prepareStatement("select * from `book` where `bno`=?");
            pstmt.setInt(1, bno);
            rs = pstmt.executeQuery();
            if(rs.next()) {
                vo = new BookVO();
                vo.setBno(rs.getInt(1));
                vo.setBname(rs.getString(2));
                vo.setBpub(rs.getString(3));
                vo.setBpric(rs.getInt(4));
            }
            close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return vo;
    }

    public List<BookVO> selectBooks() {
        List<BookVO> books = new ArrayList<>();
        try {
            conn = getConnection();
            stmt = conn.createStatement();
            rs = stmt.executeQuery("select * from `book`");
            while(rs.next()) {
                BookVO vo = new BookVO();
                vo.setBno(rs.getInt(1));
                vo.setBname(rs.getString(2));
                vo.setBpub(rs.getString(3));
                vo.setBpric(rs.getInt(4));
                books.add(vo);
            }
            close();
        } catch (Exception e) {
            e.printStackTrace();
        }
        return books;
    }

    public void updateBook(BookVO vo) {
        try {
            conn = getConnection();
            pstmt = conn.prepareStatement("update `book` set `bname`=?, `bpub`=?, `bpric`=? where `bno`=?");
            pstmt.setString(1, vo.getBname());
            pstmt.setString(2, vo.getBpub());
            pstmt.setInt(3, vo.getBpric());
            pstmt.setInt(4, vo.getBno());
            pstmt.executeUpdate();
            close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    public void deleteBook(int bno) {
        try {
            conn = getConnection();
            pstmt = conn.prepareStatement("delete from `book` where `bno`=?");
            pstmt.setInt(1, bno);
            pstmt.executeUpdate();
            close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Q 4 - book 실행

book 등록 동작

도서번호	도서명	출판사	가격	관리
2	축구의 이해	대한미디어	11001	수정 삭제
3	골프바이블	대한미디어	35000	수정 삭제
4	장발장	장발장	11001	수정 삭제

book 목록 동작

도서번호	도서명	출판사	가격	관리
2	축구의 이해	대한미디어	11001	수정 삭제
3	골프바이블	대한미디어	35000	수정 삭제

book 수정 동작

도서번호	도서명	출판사	가격	관리
2	축구의 이해	대한미디어	11000	수정 삭제
3	골프바이블	대한미디어	35000	수정 삭제
4	장발장	장발장	11001	수정 삭제

book 삭제 동작

도서번호	도서명	출판사	가격	관리
3	골프바이블	대한미디어	35000	수정 삭제
4	장발장	장발장	11001	수정 삭제

Q 4 - customer 실행

customer 등록 동작

Customer-list

localhost:8080/Bookstore2/c...

고객 목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
3	장미	대한민국 강원도맨체스터	010-1234-1003	수정 삭제
5	장발장	영국	010-1234-1234	수정 삭제
6	설록홀즈	영국	010-1234-1234	수정 삭제

customer 목록 동작

Customer-list

localhost:8080/Boo...

고객 목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
3	장미	대한민국 강원도맨체스터	010-1234-1003	수정 삭제
5	장발장	영국	010-1234-1234	수정 삭제

customer 수정 동작

Customer-list

localhost:8080/Bookstore2/c...

고객 목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
3	장미	대한민국 강원도맨체스터	010-1234-1333	수정 삭제
5	장발장	영국	010-1234-1234	수정 삭제
6	설록홀즈	영국	010-1234-1234	수정 삭제

customer 삭제 동작

Customer-list

localhost:8080/Bookstore2/c...

고객 목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
5	장발장	영국	010-1234-1234	수정 삭제
6	설록홀즈	영국	010-1234-1234	수정 삭제