



#7 통합 구현

#박진휘

index

01

프로젝트 생성

02

화면 구현

03

기능 구현

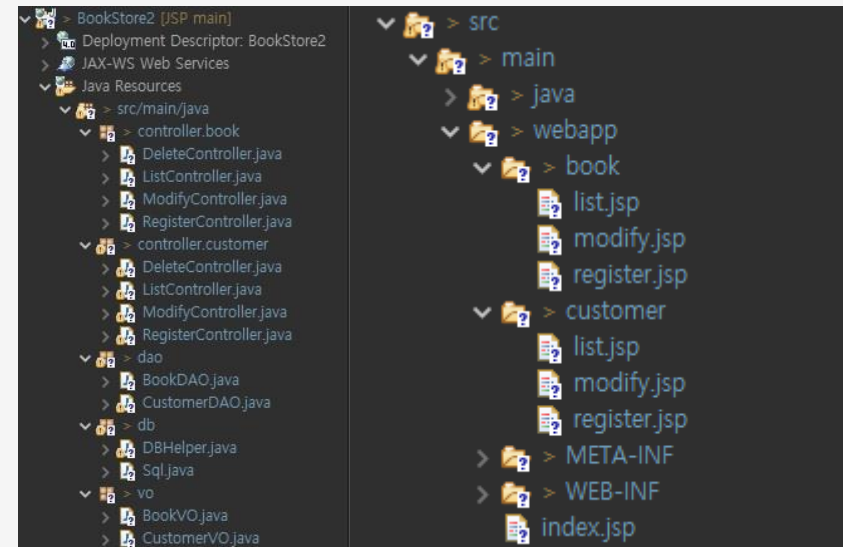
04

실행

01

프로젝트 생성

프로젝트명 : BookStore2
WAS : Tomcat 9
DB : java1_bookstore
개발도구 : Eclipse, Workbench



〇〇〇

```

1 <%@ page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"%>
2 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <meta charset="UTF-8">
7     <title>BookStore</title>
8   </head>
9   <body>
10    <h2>도서 목록</h2>
11    <a href="/BookStore2/">처음으로</a>
12    <a href="/BookStore2/book/register.do">도서 등록</a>
13
14    <table border="1">
15      <tr>
16        <th>도서번호</th>
17        <th>도서명</th>
18        <th>출판사</th>
19        <th>가격</th>
20        <th>관리</th>
21      </tr>
22      <c:forEach var="book" items="${books}">
23        <tr>
24          <td>${book.bookId }</td>
25          <td>${book.bookName }</td>
26          <td>${book.bookPublisher }</td>
27          <td>${book.bookPrice }</td>
28          <td>
29            <a href="/BookStore2/book/modify.do?bookId=${book.bookId }">수정</a>
30            <a href="/BookStore2/book/delete.do?bookId=${book.bookId }">삭제</a>
31          </td>
32        </tr>
33      </c:forEach>
34    </table>
35  </body>
36 </html>

```

〇〇〇

```

1 <%@ page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"%>
2 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <meta charset="UTF-8">
7     <title>BookStore</title>
8   </head>
9   <body>
10    <h2>고객 목록</h2>
11    <a href="/BookStore2/">처음으로</a>
12    <a href="/BookStore2/customer/register.do">고객 등록</a>
13
14    <table border="1">
15      <tr>
16        <th>고객번호</th>
17        <th>고객명</th>
18        <th>주소</th>
19        <th>휴대폰</th>
20        <th>관리</th>
21      </tr>
22      <c:forEach var="customer" items="${customers}">
23        <tr>
24          <td>${customer.custId }</td>
25          <td>${customer.name }</td>
26          <td>${customer.address }</td>
27          <td>${customer.phone }</td>
28          <td>
29            <a href="/BookStore2/customer/modify.do?custId=${customer.custId }">수정</a>
30            <a href="/BookStore2/customer/delete.do?custId=${customer.custId }">삭제</a>
31          </td>
32        </tr>
33      </c:forEach>
34    </table>
35  </body>
36 </html>

```

○○○

```

1 <%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <meta charset="UTF-8">
6     <title>BookStore</title>
7   </head>
8   <body>
9     <h2>도서등록</h2>
10    <a href="/BookStore2/">처음으로</a>
11    <a href="/BookStore2/book/list.do">도서목록</a>
12
13    <form action="/BookStore2/book/register.do" method="post">
14      <table border="1">
15        <tr>
16          <td>도서번호</td>
17          <td><input type="text" name="bookId"/></td>
18        </tr>
19        <tr>
20          <td>도서명</td>
21          <td><input type="text" name="bookName"/></td>
22        </tr>
23        <tr>
24          <td>출판사</td>
25          <td><input type="text" name="publisher"/></td>
26        </tr>
27        <tr>
28          <td>가격</td>
29          <td><input type="text" name="price"/></td>
30        </tr>
31        <tr>
32          <td colspan="2" align="right"><input type="submit" value="등록"/></td>
33        </tr>
34      </table>
35    </form>
36  </body>
37 </html>

```

○○○

```

1 <%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4   <head>
5     <meta charset="UTF-8">
6     <title>BookStore</title>
7   </head>
8   <body>
9     <h2>고객등록</h2>
10    <a href="/BookStore2/">처음으로</a>
11    <a href="/BookStore2/customer/list.do">고객목록</a>
12
13    <form action="/BookStore2/customer/register.do" method="post">
14      <table border="1">
15        <tr>
16          <td>고객명</td>
17          <td><input type="text" name="name"/></td>
18        </tr>
19        <tr>
20          <td>주소</td>
21          <td><input type="text" name="address"/></td>
22        </tr>
23        <tr>
24          <td>휴대폰</td>
25          <td><input type="text" name="phone"/></td>
26        </tr>
27        <tr>
28          <td colspan="2" align="right"><input type="submit" value="등록"/></td>
29        </tr>
30      </table>
31    </form>
32  </body>
33 </html>

```

02

Modify.jsp

〇〇〇

```

1 <%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <meta charset="UTF-8">
7     <title>BookStore</title>
8   </head>
9   <body>
10    <h2>도서 수정</h2>
11    <a href="/BookStore2/">처음으로</a>
12    <a href="/BookStore2/book/list.do">도서 목록</a>
13
14    <form action="/BookStore2/book/modify.do" method="post">
15      <table border="1">
16        <tr>
17          <td>도서번호</td>
18          <td><input type="text" name="bookId" readonly value="${vo.bookId }"/></td>
19        </tr>
20        <tr>
21          <td>도서명</td>
22          <td><input type="text" name="bookName" value="${vo.bookName }"/></td>
23        </tr>
24        <tr>
25          <td>출판사</td>
26          <td><input type="text" name="publisher" value="${vo.bookPublisher }"/></td>
27        </tr>
28        <tr>
29          <td>가격</td>
30          <td><input type="text" name="price" value="${vo.bookPrice }"/></td>
31        </tr>
32        <tr>
33          <td colspan="2" align="right"><input type="submit" value="수정"/></td>
34        </tr>
35      </table>
36    </form>
37  </body>
38 </html>

```

〇〇〇

```

1 <%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
3 <!DOCTYPE html>
4 <html>
5   <head>
6     <meta charset="UTF-8">
7     <title>BookStore</title>
8   </head>
9   <body>
10    <h2>고객 수정</h2>
11    <a href="/BookStore2/">처음으로</a>
12    <a href="/BookStore2/customer/list.do">고객 목록</a>
13
14    <form action="/BookStore2/customer/modify.do" method="post">
15      <table border="1">
16        <tr>
17          <td>고객번호</td>
18          <td><input type="text" name="custId" readonly value="${vo.custId }"/></td>
19        </tr>
20        <tr>
21          <td>고객명</td>
22          <td><input type="text" name="name" value="${vo.name }"/></td>
23        </tr>
24        <tr>
25          <td>주소</td>
26          <td><input type="text" name="address" value="${vo.address }"/></td>
27        </tr>
28        <tr>
29          <td>휴대폰</td>
30          <td><input type="text" name="phone" value="${vo.phone }"/></td>
31        </tr>
32        <tr>
33          <td colspan="2" align="right"><input type="submit" value="수정"/></td>
34        </tr>
35      </table>
36    </form>
37  </body>
38 </html>

```

03

DBHelper, Sql, index.jsp

〇〇〇

```

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

```

〇〇〇

```

1 <%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
2 <!DOCTYPE html>
3 <html>
4     <head>
5         <meta charset="UTF-8">
6         <title>BookStore</title>
7     </head>
8     <body>
9         <h2>BookStore</h2>
10        <a href="/BookStore2/book/list.do">도서목록</a>
11        <a href="/BookStore2/customer/list.do">고객목록</a>
12    </body>
13 </html>

```

〇〇〇

```

1 package db;
2
3 public class Sql {
4
5     public static final String SELECT_BOOKS = "select * from `book`";
6
7     public static final String SELECT_BOOK = "select * from `book` where `bookId`=?";
8
9     public static final String SELECT_CUSTOMERS = "select * from `customer`";
10
11     public static final String SELECT_CUSTOMER = "select * from `customer` where `custId`=?";
12
13     public static final String INSERT_BOOK = "insert into `book` value (?, ?, ?, ?)";
14
15     public static final String INSERT_CUSTOMER = "insert into `customer` (`name`, `address`, `phone`) value (?, ?, ?)";
16
17     public static final String UPDATE_BOOK = "update `book` set `bookName`=?, `publisher`=?, `price`=? where `bookId`=?";
18
19     public static final String UPDATE_CUSTOMER = "update `customer` set `name`=?, `address`=?, `phone`=? where `custId`=?";
20
21     public static final String DELETE_BOOK = "delete from `book` where `bookId`=?";
22
23     public static final String DELETE_CUSTOMER = "delete from `customer` where `custId`=?";
24
25 }

```


03

VO, DAO

〇〇〇

```

1 package vo;
2
3 public class CustomerVO {
4
5     private int custId;
6     private String name;
7     private String address;
8     private String phone;
9
10    public int getCustId() {
11        return custId;
12    }
13    public void setCustId(int custId) {
14        this.custId = custId;
15    }
16    public void setCustId(String custId) {
17        this.custId = Integer.parseInt(custId);
18    }
19    public String getName() {
20        return name;
21    }
22    public void setName(String name) {
23        this.name = name;
24    }
25    public String getAddress() {
26        return address;
27    }
28    public void setAddress(String address) {
29        this.address = address;
30    }
31    public String getPhone() {
32        return phone;
33    }
34    public void setPhone(String phone) {
35        this.phone = phone;
36    }
37
38 }

```

〇〇〇

```

1 package vo;
2
3 public class BookVO {
4
5     private String bookId;
6     private String bookName;
7     private String bookPublisher;
8     private String bookPrice;
9
10    public String getBookId() {
11        return bookId;
12    }
13    public void setBookId(String bookId) {
14        this.bookId = bookId;
15    }
16    public String getBookName() {
17        return bookName;
18    }
19    public void setBookName(String bookName) {
20        this.bookName = bookName;
21    }
22    public String getBookPublisher() {
23        return bookPublisher;
24    }
25    public void setBookPublisher(String bookPublisher) {
26        this.bookPublisher = bookPublisher;
27    }
28    public String getBookPrice() {
29        return bookPrice;
30    }
31    public void setBookPrice(String bookPrice) {
32        this.bookPrice = bookPrice;
33    }
34
35 }

```

〇〇〇

```

1 package dao;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import db.DBHelper;
7 import db.Sql;
8 import vo.BookVO;
9 import vo.CustomerVO;
10
11 public class CustomerDAO extends DBHelper {
12
13     //단일 객체
14     private static CustomerDAO instance = new CustomerDAO();
15     public static CustomerDAO getInstance() {
16         return instance;
17     }
18     private CustomerDAO () {}
19
20     public void insertCustomer(CustomerVO vo) {
21         try {
22             conn = getConnection();
23             pstmt = conn.prepareStatement(Sql.INSERT_CUSTOMER);
24             pstmt.setString(1, vo.getName());
25             pstmt.setString(2, vo.getAddress());
26             pstmt.setString(3, vo.getPhone());
27             pstmt.executeUpdate();
28             close();
29         } catch (Exception e) {
30             e.printStackTrace();
31         }
32     }
33     public CustomerVO selectCustomer(String custId) {
34         CustomerVO vo = null;
35         try {
36             conn = getConnection();
37             pstmt = conn.prepareStatement(Sql.SELECT_CUSTOMER);
38             pstmt.setString(1, custId);
39             rs = pstmt.executeQuery();
40             if(rs.next()) {
41                 vo = new CustomerVO();
42                 vo.setCustId(rs.getString(1));
43                 vo.setName(rs.getString(2));
44                 vo.setAddress(rs.getString(3));
45                 vo.setPhone(rs.getString(4));
46             }
47             close();
48         } catch (Exception e) {
49             e.printStackTrace();
50         }
51         return vo;
52     }
53     public List<CustomerVO> selectCustomers() {
54         List<CustomerVO> customers = new ArrayList<>();
55         try {
56             conn = getConnection();
57             pstmt = conn.prepareStatement(Sql.SELECT_CUSTOMERS);
58             rs = pstmt.executeQuery();
59             while(rs.next()) {
60                 CustomerVO cust = new CustomerVO();
61                 cust.setCustId(rs.getInt(1));
62                 cust.setName(rs.getString(2));
63                 cust.setAddress(rs.getString(3));
64                 cust.setPhone(rs.getString(4));
65                 customers.add(cust);
66             }
67             close();
68         } catch (Exception e) {
69             e.printStackTrace();
70         }
71         return customers;
72     }
73     public void updateCustomer(CustomerVO vo) {
74         try {
75             conn = getConnection();
76             pstmt = conn.prepareStatement(Sql.UPDATE_CUSTOMER);
77             pstmt.setString(1, vo.getName());
78             pstmt.setString(2, vo.getAddress());
79             pstmt.setString(3, vo.getPhone());
80             pstmt.setInt(4, vo.getCustId());
81             pstmt.executeUpdate();
82             close();
83         } catch (Exception e) {
84             e.printStackTrace();
85         }
86     }
87     public void deleteCustomer(String custId) {
88         try {
89             conn = getConnection();
90             pstmt = conn.prepareStatement(Sql.DELETE_CUSTOMER);
91             pstmt.setString(1, custId);
92             pstmt.executeUpdate();
93             close();
94         } catch (Exception e) {
95             e.printStackTrace();
96         }
97     }
98 }
99

```

〇〇〇

```

1 package dao;
2
3 import java.util.ArrayList;
4 import java.util.List;
5
6 import db.DBHelper;
7 import db.Sql;
8 import vo.BookVO;
9 import vo.CustomerVO;
10
11 public class BookDAO extends DBHelper {
12
13     //단일 객체
14     private static BookDAO instance = new BookDAO();
15     public static BookDAO getInstance() {
16         return instance;
17     }
18     private BookDAO () {}
19
20     public void insertBook(BookVO vo) {
21         try {
22             conn = getConnection();
23             pstmt = conn.prepareStatement(Sql.INSERT_BOOK);
24             pstmt.setString(1, vo.getBookId());
25             pstmt.setString(2, vo.getBookName());
26             pstmt.setString(3, vo.getBookPublisher());
27             pstmt.setString(4, vo.getBookPrice());
28             pstmt.executeUpdate();
29             close();
30         } catch (Exception e) {
31             e.printStackTrace();
32         }
33     }
34     public BookVO selectBook(String bookId) {
35         BookVO vo = null;
36         try {
37             conn = getConnection();
38             pstmt = conn.prepareStatement(Sql.SELECT_BOOK);
39             pstmt.setString(1, bookId);
40             rs = pstmt.executeQuery();
41             if(rs.next()) {
42                 vo = new BookVO();
43                 vo.setBookId(rs.getString(1));
44                 vo.setBookName(rs.getString(2));
45                 vo.setBookPublisher(rs.getString(3));
46                 vo.setBookPrice(rs.getString(4));
47             }
48             close();
49         } catch (Exception e) {
50             e.printStackTrace();
51         }
52         return vo;
53     }
54     public List<BookVO> selectBooks() {
55         List<BookVO> books = new ArrayList<>();
56         try {
57             conn = getConnection();
58             pstmt = conn.prepareStatement(Sql.SELECT_BOOKS);
59             rs = pstmt.executeQuery();
60             while(rs.next()) {
61                 BookVO book = new BookVO();
62                 book.setBookId(rs.getString(1));
63                 book.setBookName(rs.getString(2));
64                 book.setBookPublisher(rs.getString(3));
65                 book.setBookPrice(rs.getString(4));
66                 books.add(book);
67             }
68             close();
69         } catch (Exception e) {
70             e.printStackTrace();
71         }
72         return books;
73     }
74     public void updateBook(BookVO vo) {
75         try {
76             conn = getConnection();
77             pstmt = conn.prepareStatement(Sql.UPDATE_BOOK);
78             pstmt.setString(1, vo.getBookName());
79             pstmt.setString(2, vo.getBookPublisher());
80             pstmt.setString(3, vo.getBookPrice());
81             pstmt.setString(4, vo.getBookId());
82             pstmt.executeUpdate();
83             close();
84         } catch (Exception e) {
85             e.printStackTrace();
86         }
87     }
88     public void deleteBook(String bookId) {
89         try {
90             conn = getConnection();
91             pstmt = conn.prepareStatement(Sql.DELETE_BOOK);
92             pstmt.setString(1, bookId);
93             pstmt.executeUpdate();
94             close();
95         } catch (Exception e) {
96             e.printStackTrace();
97         }
98     }
99 }

```


03

ListController

```
○○○
1 package controller.book;
2
3 import java.io.IOException;
4 import java.util.List;
5
6 import javax.servlet.RequestDispatcher;
7 import javax.servlet.ServletException;
8 import javax.servlet.annotation.WebServlet;
9 import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12
13 import dao.BookDAO;
14 import vo.BookVO;
15
16 @WebServlet("/book/list.do")
17 public class ListController extends HttpServlet {
18
19     private static final long serialVersionUID = 1L;
20
21     @Override
22     protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
23         List<BookVO> books = BookDAO.getInstance().selectBooks();
24         req.setAttribute("books", books);
25
26         RequestDispatcher dispatcher = req.getRequestDispatcher("/book/list.jsp");
27         dispatcher.forward(req, resp);
28     }
29
30 }
```

```
○○○
1 package controller.customer;
2
3 import java.io.IOException;
4 import java.util.List;
5
6 import javax.servlet.RequestDispatcher;
7 import javax.servlet.ServletException;
8 import javax.servlet.annotation.WebServlet;
9 import javax.servlet.http.HttpServlet;
10 import javax.servlet.http.HttpServletRequest;
11 import javax.servlet.http.HttpServletResponse;
12
13 import dao.BookDAO;
14 import dao.CustomerDAO;
15 import vo.BookVO;
16 import vo.CustomerVO;
17
18 @WebServlet("/customer/list.do")
19 public class ListController extends HttpServlet {
20
21     private static final long serialVersionUID = 1L;
22
23     @Override
24     protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
25         List<CustomerVO> customers = CustomerDAO.getInstance().selectCustomers();
26         req.setAttribute("customers", customers);
27
28         RequestDispatcher dispatcher = req.getRequestDispatcher("/customer/list.jsp");
29         dispatcher.forward(req, resp);
30     }
31
32 }
```

03

RegisterController

○○○

```
1 package controller.book;
2
3 import java.io.IOException;
4
5 import javax.servlet.RequestDispatcher;
6 import javax.servlet.ServletException;
7 import javax.servlet.annotation.WebServlet;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11
12 import dao.BookDAO;
13 import vo.BookVO;
14
15 @WebServlet("/book/register.do")
16 public class RegisterController extends HttpServlet {
17
18     private static final long serialVersionUID = 1L;
19
20     @Override
21     protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
22         RequestDispatcher dispatcher = req.getRequestDispatcher("/book/register.jsp");
23         dispatcher.forward(req, resp);
24     }
25     @Override
26     protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
27
28         String bookId = req.getParameter("bookId");
29         String bookName = req.getParameter("bookName");
30         String bookPublisher = req.getParameter("publisher");
31         String bookPrice = req.getParameter("price");
32
33         BookVO vo = new BookVO();
34         vo.setBookId(bookId);
35         vo.setBookName(bookName);
36         vo.setBookPublisher(bookPublisher);
37         vo.setBookPrice(bookPrice);
38
39         BookDAO.getInstance().insertBook(vo);
40
41         resp.sendRedirect("/BookStore2/book/list.do");
42     }
43 }
```

○○○

```
1 package controller.customer;
2
3 import java.io.IOException;
4
5 import javax.servlet.RequestDispatcher;
6 import javax.servlet.ServletException;
7 import javax.servlet.annotation.WebServlet;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11
12 import dao.BookDAO;
13 import dao.CustomerDAO;
14 import vo.BookVO;
15 import vo.CustomerVO;
16
17 @WebServlet("/customer/register.do")
18 public class RegisterController extends HttpServlet {
19
20     private static final long serialVersionUID = 1L;
21
22     @Override
23     protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
24         RequestDispatcher dispatcher = req.getRequestDispatcher("/customer/register.jsp");
25         dispatcher.forward(req, resp);
26     }
27     @Override
28     protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
29
30         String name = req.getParameter("name");
31         String address = req.getParameter("address");
32         String phone = req.getParameter("phone");
33
34         CustomerVO vo = new CustomerVO();
35         vo.setName(name);
36         vo.setAddress(address);
37         vo.setPhone(phone);
38
39         CustomerDAO.getInstance().insertCustomer(vo);
40
41         resp.sendRedirect("/BookStore2/customer/list.do");
42     }
43 }
```

03

ModifyController

```

○○○
1 package controller.book;
2
3 import java.io.IOException;
4
5 import javax.servlet.RequestDispatcher;
6 import javax.servlet.ServletException;
7 import javax.servlet.annotation.WebServlet;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11
12 import dao.BookDAO;
13 import vo.BookVO;
14
15 @WebServlet("/book/modify.do")
16 public class ModifyController extends HttpServlet {
17
18     private static final long serialVersionUID = 1L;
19
20     @Override
21     protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
22         String bookId = req.getParameter("bookId");
23         BookVO vo = BookDAO.getInstance().selectBook(bookId);
24         req.setAttribute("vo", vo);
25
26         RequestDispatcher dispatcher = req.getRequestDispatcher("/book/modify.jsp");
27         dispatcher.forward(req, resp);
28     }
29     @Override
30     protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
31
32         String bookId = req.getParameter("bookId");
33         String bookName = req.getParameter("bookName");
34         String bookPublisher = req.getParameter("publisher");
35         String bookPrice = req.getParameter("price");
36
37         BookVO vo = new BookVO();
38         vo.setBookId(bookId);
39         vo.setBookName(bookName);
40         vo.setBookPublisher(bookPublisher);
41         vo.setBookPrice(bookPrice);
42
43         BookDAO.getInstance().updateBook(vo);
44
45         resp.sendRedirect("/BookStore2/book/list.do");
46     }
47 }
48

```

```

○○○
1 package controller.customer;
2
3 import java.io.IOException;
4
5 import javax.servlet.RequestDispatcher;
6 import javax.servlet.ServletException;
7 import javax.servlet.annotation.WebServlet;
8 import javax.servlet.http.HttpServlet;
9 import javax.servlet.http.HttpServletRequest;
10 import javax.servlet.http.HttpServletResponse;
11
12 import dao.BookDAO;
13 import dao.CustomerDAO;
14 import vo.BookVO;
15 import vo.CustomerVO;
16
17 @WebServlet("/customer/modify.do")
18 public class ModifyController extends HttpServlet {
19
20     private static final long serialVersionUID = 1L;
21
22     @Override
23     protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
24         String custId = req.getParameter("custId");
25         CustomerVO vo = CustomerDAO.getInstance().selectCustomer(custId);
26         req.setAttribute("vo", vo);
27
28         RequestDispatcher dispatcher = req.getRequestDispatcher("/customer/modify.jsp");
29         dispatcher.forward(req, resp);
30     }
31     @Override
32     protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
33
34         String custId = req.getParameter("custId");
35         String name = req.getParameter("name");
36         String address = req.getParameter("address");
37         String phone = req.getParameter("phone");
38
39         CustomerVO vo = new CustomerVO();
40         vo.setCustId(custId);
41         vo.setName(name);
42         vo.setAddress(address);
43         vo.setPhone(phone);
44
45         CustomerDAO.getInstance().updateCustomer(vo);
46
47         resp.sendRedirect("/BookStore2/customer/list.do");
48     }
49 }
50

```

03

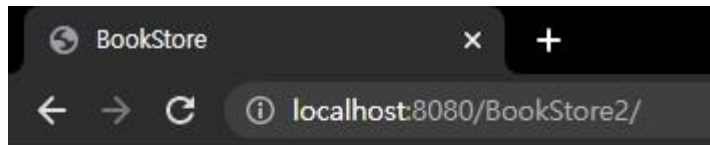
DeleteController

```
○○○
1 package controller.book;
2
3 import java.io.IOException;
4
5 import javax.servlet.ServletException;
6 import javax.servlet.annotation.WebServlet;
7 import javax.servlet.http.HttpServlet;
8 import javax.servlet.http.HttpServletRequest;
9 import javax.servlet.http.HttpServletResponse;
10
11 import dao.BookDAO;
12
13 @WebServlet("/book/delete.do")
14 public class DeleteController extends HttpServlet{
15
16     private static final long serialVersionUID = 1L;
17
18     @Override
19     protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
20         String bookId = req.getParameter("bookId");
21         BookDAO.getInstance().deleteBook(bookId);
22
23         resp.sendRedirect("/BookStore2/book/list.do");
24     }
25 }
```

```
○○○
1 package controller.customer;
2
3 import java.io.IOException;
4
5 import javax.servlet.ServletException;
6 import javax.servlet.annotation.WebServlet;
7 import javax.servlet.http.HttpServlet;
8 import javax.servlet.http.HttpServletRequest;
9 import javax.servlet.http.HttpServletResponse;
10
11 import dao.BookDAO;
12 import dao.CustomerDAO;
13
14 @WebServlet("/customer/delete.do")
15 public class DeleteController extends HttpServlet{
16
17     private static final long serialVersionUID = 1L;
18
19     @Override
20     protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
21         String custId = req.getParameter("custId");
22         CustomerDAO.getInstance().deleteCustomer(custId);
23
24         resp.sendRedirect("/BookStore2/customer/list.do");
25     }
26 }
```

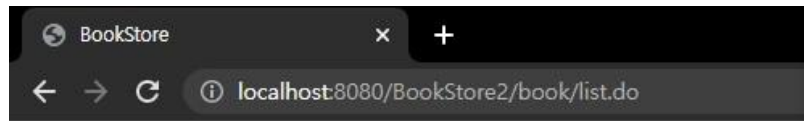
04

실행



BookStore

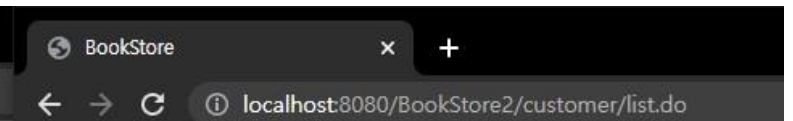
[도서목록](#) [고객목록](#)



도서목록

[처음으로](#) [도서등록](#)

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	수정 삭제
2	축구아는 여자	나무수	13000	수정 삭제
3	축구의 이해	대한미디어	22000	수정 삭제
4	골프 바이블	대한미디어	35000	수정 삭제
5	피겨 교본	굿스포츠	8000	수정 삭제
6	역도 단계별기술	굿스포츠	6000	수정 삭제
7	농구의 추억	이상미디어	20000	수정 삭제
8	농구를 부탁해	이상미디어	13000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Chmpions	Pearson	13000	수정 삭제
11	스포츠의학	한솔의학서적		수정 삭제



고객목록

[처음으로](#) [고객등록](#)

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스터	000-5000-0001	수정 삭제
2	김연아	대한민국 서울	000-6000-0001	수정 삭제
3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
4	추신수	미국 클리블랜드	000-8000-0001	수정 삭제
6	박세리	대한민국 대전		수정 삭제

리스트 - DB 값 출력

04

실행

BookStore

localhost:8080/BookStore2/book/register.do

도서등록

처음으로 도서목록

도서번호	101
도서명	테스트
출판사	테스트사
가격	100000
<div>등록</div>	

BookStore

localhost:8080/BookStore2/c

고객등록

처음으로 고객목록

고객명	테스터
주소	부산시
휴대폰	000-1111-2222
<div>등록</div>	

8	농구를 부탁해	이상미디어	13000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Chmpions	Pearson	13000	수정 삭제
11	스포츠의학	한솔의학서적		수정 삭제
101	테스트	테스트사	100000	수정 삭제

4	추신수	미국 클리블랜드	000-8000-0001	수정 삭제
6	박세리	대한민국 대전		수정 삭제
10	테스터	부산시	000-1111-2222	수정 삭제

등록 - 새로운 데이터 값 등록

04

실행

BookStore

localhost:8080/BookStore2/book/modify.do?bookId=101

도서수정

처음으로 도서목록

도서번호	101
도서명	테스트
출판사	테스트사
가격	100000
<input type="button" value="수정"/>	

BookStore

localhost:8080/BookStore2/cu

고객수정

처음으로 고객목록

고객번호	10
고객명	테스터
주소	부산시
휴대폰	000-1111-2222
<input type="button" value="수정"/>	

수정 - 기존의 데이터 값 출력

3	축구의 이해	대한미디어	22000	수정 삭제
4	골프 바이블	대한미디어	35000	수정 삭제
5	피겨 교본	굿스포츠	8000	수정 삭제
6	역도 단계별기술	굿스포츠	6000	수정 삭제
7	농구의 추억	이상미디어	20000	수정 삭제
8	농구를 부탁해	이상미디어	13000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Chmpions	Pearson	13000	수정 삭제
11	스포츠의학	한솔의학서적		수정 삭제
101	테스트1	테스트사1	20000	수정 삭제

수정된 데이터 값

5	피겨 교본	굿스포츠	8000	수정 삭제
6	역도 단계별기술	굿스포츠	6000	수정 삭제
7	농구의 추억	이상미디어	20000	수정 삭제
8	농구를 부탁해	이상미디어	13000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Chmpions	Pearson	13000	수정 삭제
11	스포츠의학	한솔의학서적		수정 삭제

데이터 삭제

3	장미란	대한민국 강원도	000-7000-0001	수정 삭제
4	추신수	미국 클리블랜드	000-8000-0001	수정 삭제
6	박세리	대한민국 대전		수정 삭제
10	테스터	부산시진구	000-1111-3333	수정 삭제

삭제 클릭

6	박세리	대한민국 대전		수정 삭제
---	-----	---------	--	-------

Thank you