




통합 구현

2022/11/18(금)

김지홍

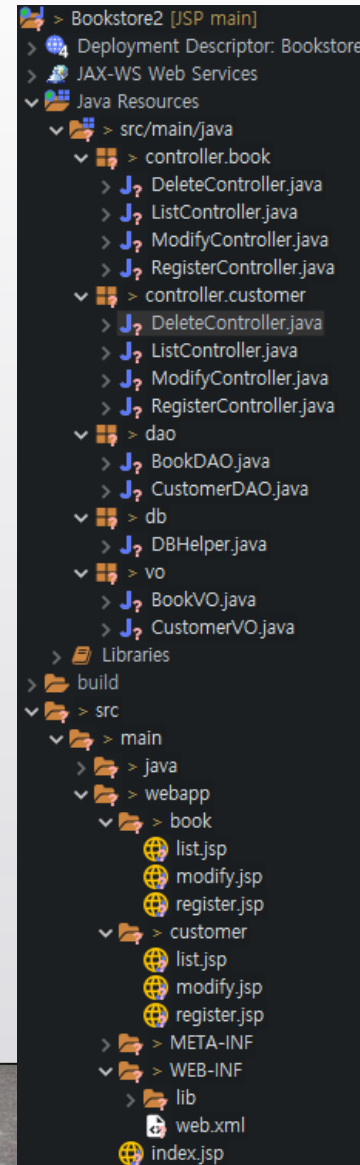


목차

1. 프로젝트 생성
2. 화면 구현
3. 기능 구현
4. 실행

문제 1. 프로젝트 생성

3



문제 2. 화면 구현

4

BookStore

[도서목록](#) [고객목록](#)

도서목록

[처음으로 도서등록](#)

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	수정 삭제
2	축구하는 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	야구의 추억	이상미디어	20000	수정 삭제
8	야구를 부탁해	이상미디어	13000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제

도서등록

[처음으로 도서목록](#)

도서번호	102
도서명	자바웹프로그래밍
출판사	프로그래밍출판사
가격	20000
<input type="button" value="등록"/>	

도서수정

[처음으로 도서목록](#)

도서번호	102
도서명	자바웹프로그래밍
출판사	프로그래밍출판사
가격	20000
<input type="button" value="수정"/>	

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
5	박세리	대한민국 부산		수정 삭제
6	이단계사	광주	010-1234	수정 삭제

고객등록

[처음으로 고객목록](#)

고객명	류현진
주소	클리블랜드
휴대폰	011-0000-0001
<input type="button" value="등록"/>	

고객수정

[처음으로 고객목록](#)

고객번호	12
고객명	류현진
주소	클리블랜드
휴대폰	011-0000-0001
<input type="button" value="수정"/>	

문제 3. 기능 구현

5

도서(Model)

```
public class BookVO {
    private int bookId;
    private String bookName;
    private String publisher;
    private int price;

    public int getBookId() {
        return bookId;
    }
    public void setBookId(int bookId) {
        this.bookId = bookId;
    }
    public void setBookId(String bookId) {
        this.bookId = Integer.parseInt(bookId);
    }
    public String getBookName() {
        return bookName;
    }
    public void setBookName(String bookName) {
        this.bookName = bookName;
    }
    public String getPublisher() {
        return publisher;
    }
    public void setPublisher(String publisher) {
        this.publisher = publisher;
    }
    public int getPrice() {
        return price;
    }
    public void setPrice(int price) {
        this.price = price;
    }
    public void setPrice(String price) {
        this.price = Integer.parseInt(price);
    }
}
```

고객(Model)

```
public class CustomerVO {
    private int custId;
    private String name;
    private String address;
    private String phone;

    public int getCustId() {
        return custId;
    }
    public void setCustId(int custId) {
        this.custId = custId;
    }
    public void setCustId(String custId) {
        this.custId = Integer.parseInt(custId);
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getAddress() {
        return address;
    }
    public void setAddress(String address) {
        this.address = address;
    }
    public String getPhone() {
        return phone;
    }
    public void setPhone(String phone) {
        this.phone = phone;
    }
}
```


문제 3. 기능 구현

6

도서목록

Controller

```
@WebServlet("/book/list.do")
public class ListController extends HttpServlet{
    private static final long serialVersionUID = 1L;

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException {
        req.setAttribute("books", BookDAO.getInstance().selectBooks());
        RequestDispatcher dispatcher = req.getRequestDispatcher("/book/list.jsp");
        dispatcher.forward(req, resp);
    }
}
```

DAO

```
public List<BookVO> selectBooks() {
    List<BookVO> books = null;
    try {
        conn = getConnection();
        stmt = conn.createStatement();
        rs = stmt.executeQuery("SELECT * FROM `book`");
        books = new ArrayList<>();
        while(rs.next()) {
            BookVO book = new BookVO();
            book.setBookId(rs.getInt(1));
            book.setBookName(rs.getString(2));
            book.setPublisher(rs.getString(3));
            book.setPrice(rs.getInt(4));
            books.add(book);
        }
        close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return books;
}
```

View(JSTL)

```
<c:forEach var="book" items="${requestScope.books}">
<tr>
    <td>${book.bookId}</td>
    <td>${book.bookName}</td>
    <td>${book.publisher}</td>
    <td>${book.price}</td>
    <td>
        <a href="/modify.do?bookId=${book.bookId}">수정</a>
        <a href="/delete.do?bookId=${book.bookId}">삭제</a>
    </td>
</tr>
</c:forEach>
```

고객목록

Controller

```
@WebServlet("/customer/list.do")
public class ListController extends HttpServlet{
    private static final long serialVersionUID = 1L;

    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException {
        req.setAttribute("customers", CustomerDAO.getInstance().selectCustomers());
        RequestDispatcher dispatcher = req.getRequestDispatcher("/customer/list.jsp");
        dispatcher.forward(req, resp);
    }
}
```

DAO

```
public List<CustomerVO> selectCustomers() {
    List<CustomerVO> customers = null;
    try {
        conn = getConnection();
        stmt = conn.createStatement();
        rs = stmt.executeQuery("SELECT * FROM `customer`");
        customers = new ArrayList<>();
        while(rs.next()) {
            CustomerVO customer = new CustomerVO();
            customer.setCustId(rs.getInt(1));
            customer.setName(rs.getString(2));
            customer.setAddress(rs.getString(3));
            customer.setPhone(rs.getString(4));
            customers.add(customer);
        }
        close();
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return customers;
}
```

View(JSTL)

```
<c:forEach var="customer" items="${requestScope.customers}">
<tr>
    <td>${customer.custId}</td>
    <td>${customer.name}</td>
    <td>${customer.address}</td>
    <td>${customer.phone}</td>
    <td>
        <a href="/modify.do?custId=${customer.custId}">수정</a>
        <a href="/delete.do?custId=${customer.custId}">삭제</a>
    </td>
</tr>
</c:forEach>
```

문제 3. 기능 구현

7

도서등록

Controller

```
@Override
protected void doPost(HttpServletRequest req, HttpServletResponse res) {
    BookVO book = new BookVO();
    book.setBookId(req.getParameter("bookId"));
    book.setBookName(req.getParameter("bookName"));
    book.setPublisher(req.getParameter("publisher"));
    book.setPrice(req.getParameter("price"));
    BookDAO.getInstance().insertBook(book);

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

DAO

```
public void insertBook(BookVO book) {
    try {
        conn = getConnection();
        psmt = conn.prepareStatement("INSERT `book` SET `bookId`=?, `bookName`=?, `publisher`=?, `price`=?");
        psmt.setInt(1, book.getBookId());
        psmt.setString(2, book.getBookName());
        psmt.setString(3, book.getPublisher());
        psmt.setInt(4, book.getPrice());
        psmt.executeUpdate();
        close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

View(JSTL)

```
<form action="/register.do" method="post">
  <table border="1">
    <tr>
      <td>도서번호</td>
      <td><input type="text" name="bookId"/></td>
    </tr>
    <tr>
      <td>도서명</td>
      <td><input type="text" name="bookName"/></td>
    </tr>
    <tr>
      <td>출판사</td>
      <td><input type="text" name="publisher"/></td>
    </tr>
    <tr>
      <td>가격</td>
      <td><input type="text" name="price"/></td>
    </tr>
    <tr>
      <td colspan="2" align="right">
        <input type="submit" value="등록" />
      </td>
    </tr>
  </table>
</form>
```

고객등록

Controller

```
@Override
protected void doPost(HttpServletRequest req, HttpServletResponse res) {
    CustomerVO customer = new CustomerVO();
    customer.setName(req.getParameter("name"));
    customer.setAddress(req.getParameter("address"));
    customer.setPhone(req.getParameter("phone"));

    CustomerDAO.getInstance().insertCustomer(customer);

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

DAO

```
public void insertCustomer(CustomerVO customer) {
    try {
        conn = getConnection();
        psmt = conn.prepareStatement("INSERT `Customer` SET `name`=?, `address`=?, `phone`=?");
        psmt.setString(1, customer.getName());
        psmt.setString(2, customer.getAddress());
        psmt.setString(3, customer.getPhone());
        psmt.executeUpdate();
        close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

View(JSTL)

```
<form action="/register.do" method="post">
  <table border="1">
    <tr>
      <td>고객명</td>
      <td><input type="text" name="name"/></td>
    </tr>
    <tr>
      <td>주소</td>
      <td><input type="text" name="address"/></td>
    </tr>
    <tr>
      <td>휴대폰</td>
      <td><input type="text" name="phone"/></td>
    </tr>
    <tr>
      <td colspan="2" align="right">
        <input type="submit" value="등록" />
      </td>
    </tr>
  </table>
</form>
```

문제 3. 기능 구현

8

도서수정

Controller

```
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    req.setAttribute("book", BookDAO.getInstance().selectBook(req.getParameter("bookId")));
    RequestDispatcher dispatcher = req.getRequestDispatcher("/book/modify.jsp");
    dispatcher.forward(req, resp);
}

@Override
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    BookVO book = new BookVO();
    book.setBookId(req.getParameter("bookId"));
    book.setBookName(req.getParameter("bookName"));
    book.setPublisher(req.getParameter("publisher"));
    book.setPrice(req.getParameter("price"));
    BookDAO.getInstance().updateBook(book);
    resp.sendRedirect("/Bookstore2/book/list.do");
}
```

DAO

```
public void updateBook(BookVO book) {
    try {
        conn = getConnection();
        pstmt = conn.prepareStatement("UPDATE `book` SET `bookName`=?, `publisher`=?, `price`=? WHERE `bookId`=?");
        pstmt.setString(1, book.getBookName());
        pstmt.setString(2, book.getPublisher());
        pstmt.setInt(3, book.getPrice());
        pstmt.setInt(4, book.getBookId());
        pstmt.executeUpdate();
        close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

View(JSTL)

```
<form action="/modify.do" method="post">
<table border="1">
<tr>
<td>도서번호</td>
<td><input type="text" name="bookId" value="${book.bookId}" readonly/></td>
</tr>
<tr>
<td>도서명</td>
<td><input type="text" name="bookName" value="${book.bookName}" /></td>
</tr>
<tr>
<td>출판사</td>
<td><input type="text" name="publisher" value="${book.publisher}" /></td>
</tr>
<tr>
<td>가격</td>
<td><input type="text" name="price" value="${book.price}" /></td>
</tr>
<tr>
<td align="right" colspan="2">
<input type="submit" value="수정" />
</td>
</tr>
</table>
</form>
```

고객수정

Controller

```
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    req.setAttribute("customer", CustomerDAO.getInstance().selectCustomer(req.getParameter("custId")));
    RequestDispatcher dispatcher = req.getRequestDispatcher("/customer/modify.jsp");
    dispatcher.forward(req, resp);
}

@Override
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    CustomerVO customer = new CustomerVO();
    customer.setCustId(req.getParameter("custId"));
    customer.setName(req.getParameter("name"));
    customer.setAddress(req.getParameter("address"));
    customer.setPhone(req.getParameter("phone"));
    CustomerDAO.getInstance().updateCustomer(customer);
    resp.sendRedirect("/Bookstore2/customer/list.do");
}
```

DAO

```
public void updateCustomer(CustomerVO customer) {
    try {
        conn = getConnection();
        pstmt = conn.prepareStatement("UPDATE `customer` SET `name`=?, `address`=?, `phone`=? WHERE `custId`=?");
        pstmt.setString(1, customer.getName());
        pstmt.setString(2, customer.getAddress());
        pstmt.setString(3, customer.getPhone());
        pstmt.setInt(4, customer.getCustId());
        pstmt.executeUpdate();
        close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

View(JSTL)

```
<form action="/modify.do" method="post">
<table border="1">
<tr>
<td>고객번호</td>
<td><input type="text" name="custId" value="${customer.custId}" readonly/></td>
</tr>
<tr>
<td>고객명</td>
<td><input type="text" name="name" value="${customer.name}" /></td>
</tr>
<tr>
<td>주소</td>
<td><input type="text" name="address" value="${customer.address}" /></td>
</tr>
<tr>
<td>휴대폰</td>
<td><input type="text" name="phone" value="${customer.phone}" /></td>
</tr>
<tr>
<td align="right" colspan="2">
<input type="submit" value="수정" />
</td>
</tr>
</table>
</form>
```


문제 3. 기능 구현

9

도서삭제

Controller

```
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) {
    BookDAO.getInstance().deleteBook(req.getParameter("bookId"));
    resp.sendRedirect("/Bookstore2/book/list.do");
}
```

DAO

```
public void deleteBook(String bookId) {
    try {
        conn = getConnection();
        psmt = conn.prepareStatement("DELETE FROM `book` WHERE `bookId`=?");
        psmt.setString(1, bookId);
        psmt.executeUpdate();
        close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

고객삭제

Controller

```
@Override
protected void doGet(HttpServletRequest req, HttpServletResponse resp) {
    CustomerDAO.getInstance().deleteCustomer(req.getParameter("custId"));
    resp.sendRedirect("/Bookstore2/customer/list.do");
}
```

DAO

```
public void deleteCustomer(String custId) {
    try {
        conn = getConnection();
        psmt = conn.prepareStatement("DELETE FROM `customer` WHERE `custId`=?");
        psmt.setString(1, custId);
        psmt.executeUpdate();
        close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

Controller 에서 get, post 요청에 맞게 페이지 또는 데이터 응답

DAO 에서 DB를 이용하여 CRUD 후 데이터 또는 결과 응답

View 는 Controller에서 request에서 담은 데이터를 JSTL, EL 을 활용하여 출력

context.xml

```
<!-- java1_bookstore -->
<Resource
    name="dbcp_java1_bookstore"
    auth="Container"
    type="javax.sql.DataSource"
    driverClassName="com.mysql.cj.jdbc.Driver"
    url="jdbc:mysql://127.0.0.1:3306/java1_bookstore"
    username="root"
    password="1234"
    maxTotal="8"
    maxIdle="8"
    maxWaitMillis="3000"
/>
```

DBCP(DBHelper)

```
public class DBHelper {

    protected Connection conn = null;
    protected PreparedStatement psmt = null;
    protected Statement stmt = null;
    protected ResultSet rs = null;

    public Connection getConnection() {

        try {
            DataSource ds = (DataSource)new InitialContext().lookup("java:comp/env/dbcp_java1_bookstore");
            conn=ds.getConnection();

        }catch(Exception e) {
            e.printStackTrace();
        }
        return conn;
    }

    public void close() {
        try {
            if(rs != null) {
                rs.close();
            }
            if(stmt != null) {
                stmt.close();
            }
            if(psmt != null) {
                psmt.close();
            }
            if(conn != null) {
                conn.close();
            }
        }catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

문제 4. 실행

11

도서등록

[처음으로 도서목록](#)

도서번호	102
도서명	자바웹프로그래밍
출판사	프로그래밍출판사
가격	20000
<input type="button" value="등록"/>	

10	Olympic Champions	Pearson	13000	수정 삭제
102	자바웹프로그래밍	프로그래밍출판사	20000	수정 삭제

도서수정

[처음으로 도서목록](#)

도서번호	102
도서명	자바웹프로그래밍2
출판사	웹출판사
가격	25000
<input type="button" value="수정"/>	

10	Olympic Champions	Pearson	13000	수정 삭제
102	자바웹프로그래밍2	웹출판사	25000	수정 삭제
103	HTML/CSS	웹프로그래밍	30000	수정 삭제

10	Olympic Champions	Pearson	13000	수정 삭제
103	HTML/CSS	웹프로그래밍	30000	수정 삭제

고객등록

[처음으로 고객목록](#)

고객명	류현진
주소	클리블랜드
휴대폰	011-0000-0001
<input type="button" value="등록"/>	

6	이단계사	광주	010-1234	수정 삭제
12	류현진	클리블랜드	011-0000-0001	수정 삭제
13	손흥민	영국	011-0000-0002	수정 삭제

고객수정

[처음으로 고객목록](#)

고객번호	12
고객명	박찬호
주소	LA
휴대폰	011-1000-0001
<input type="button" value="수정"/>	

6	이단계사	광주	010-1234	수정 삭제
12	박찬호	LA	011-1000-0001	수정 삭제
13	손흥민	영국	011-0000-0002	수정 삭제

6	이단계사	광주	010-1234	수정 삭제
13	손흥민	영국	011-0000-0002	수정 삭제