

# 11. 서버 프로그램 구현

응용 프로그래밍 및 화면구현

날짜 : 2023/ 01/ 10 (화)

이름 : 김재준

# 목차

› 프로젝트 생성 및 구성

› 화면 구현

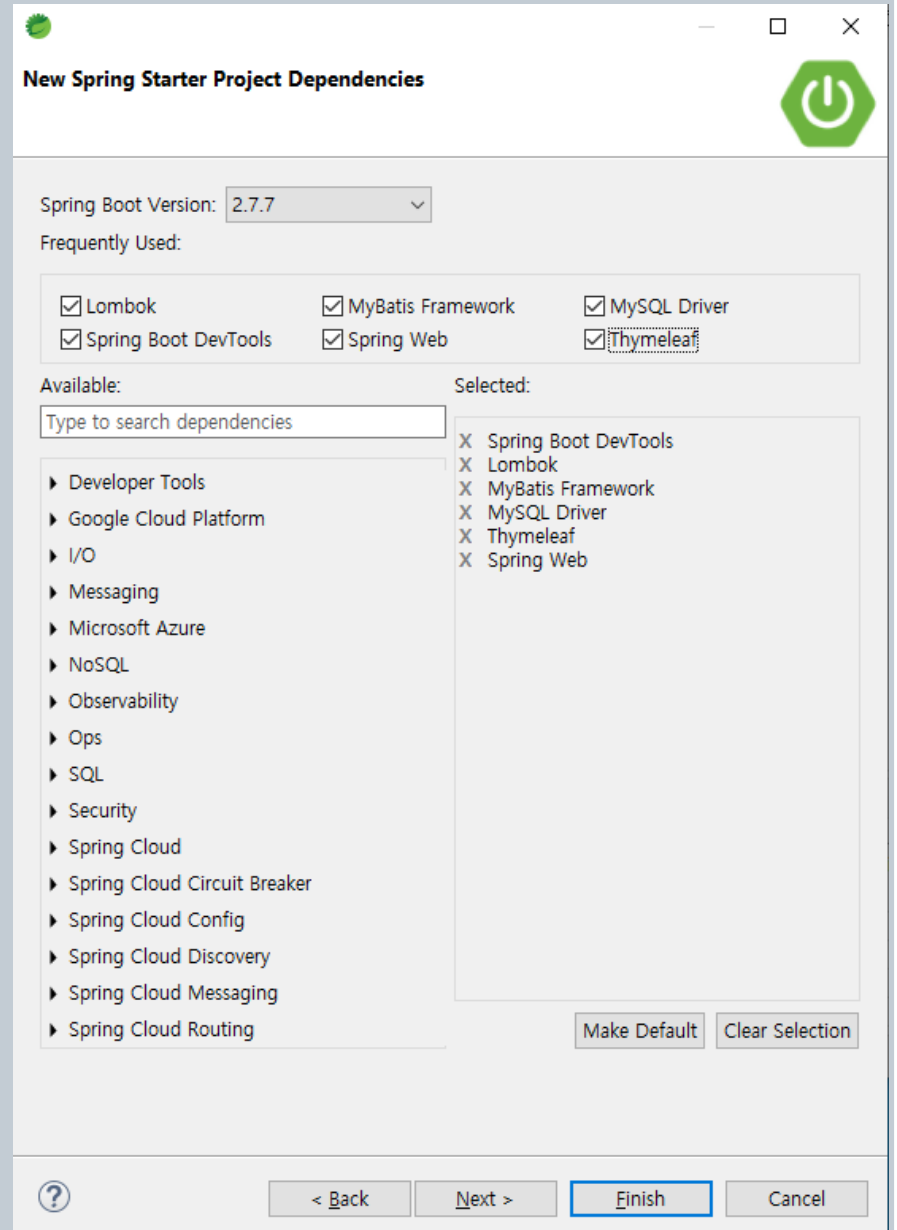
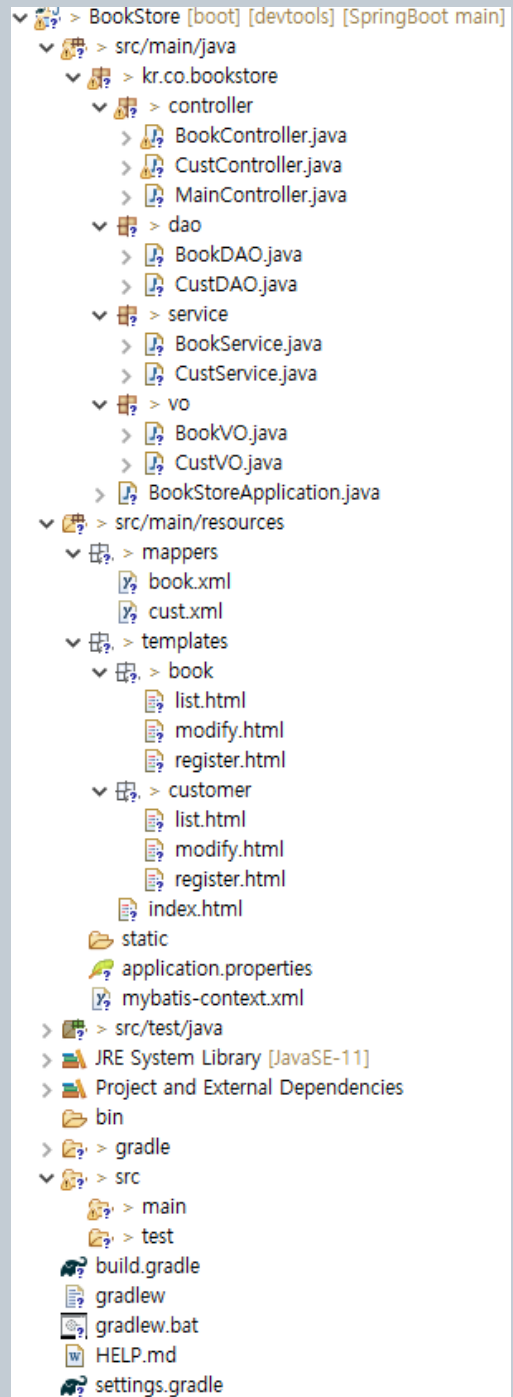
› 기능 구현

› 실행

# 과정 수행

# 1. 프로젝트 생성 및 구성

- 프로젝트 : BookStore
- WAS : Tomcat9
- DB : java2\_bookstore
- Framework : SpringBoot, Mybatis
- Tools : Eclipse, Workbench



- Application.properties

```
1 server.servlet.context-path=/BookStore
2 server.port=8080
3 spring.thymeleaf.cache=false
4
5 #Mybatis 설정
6
7 #데이터베이스 정보
8 spring.datasource.url=jdbc:mysql://127.0.0.1:3306/java2_bookstore
9 spring.datasource.username=root
10 spring.datasource.password=1234
11 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
12
13 #MyBatis Mapper 경로설정
14 mybatis.mapper-locations=classpath:mappers/**/*.xml
```

- BookStoreApplication

```
1 package kr.co.bookstore;
2
3 import org.mybatis.spring.annotation.MapperScan;
4
5
6
7 @MapperScan("kr.co.bookstore.persistence")
8 @SpringBootApplication
9 public class BookStoreApplication {
10
11     public static void main(String[] args) {
12         SpringApplication.run(BookStoreApplication.class, args);
13     }
14
15 }
16
```

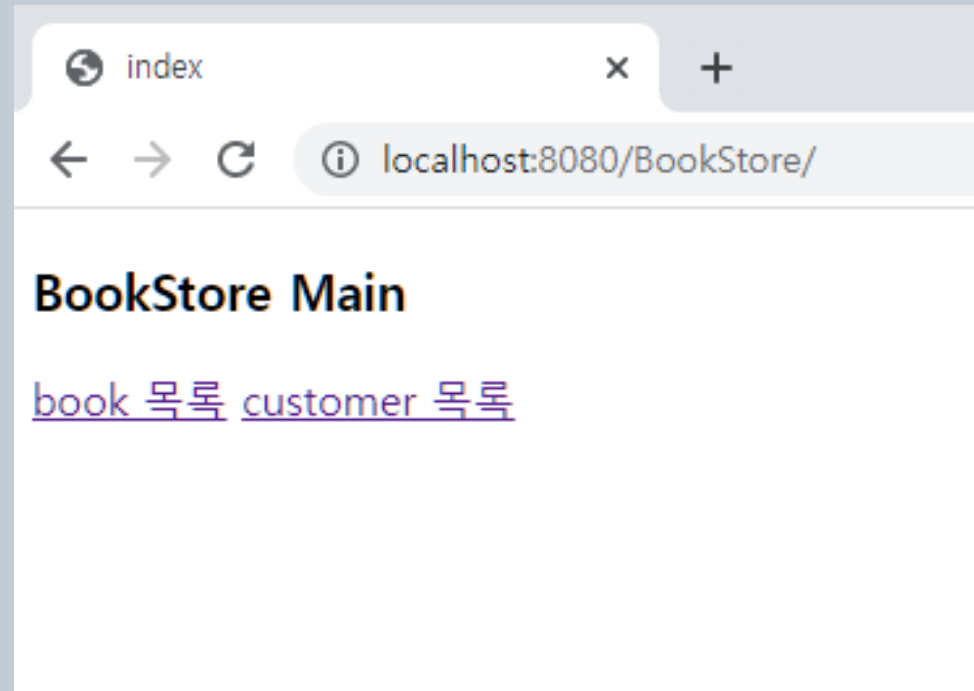
## 2. 화면 구현

- book(도서)

목록, 등록, 수정 화면 구현

- customer(고객)

목록, 등록, 수정 화면 구현



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>index</title>
6   </head>
7   <body>
8     <h3>BookStore Main</h3>
9
10    <a href="/BookStore/book/list">book 목록</a>
11    <a href="/BookStore/customer/list">customer 목록</a>
12
13  </body>
14 </html>
```

## • book, customer list 코드, 화면

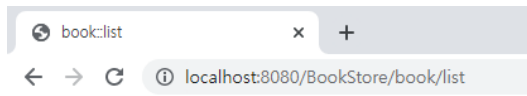
### • book list

```

1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <meta charset="UTF-8">
5     <title>book::list</title>
6   </head>
7   <body>
8     <h3>book 목록</h3>
9     <a href="/BookStore">BookStore 메인</a>
10    <a href="/BookStore/book/register">book 등록</a>
11
12    <table border="1">
13      <tr>
14        <th>도서ID</th>
15        <th>도서명</th>
16        <th>출판사</th>
17        <th>가격</th>
18        <th>관리</th>
19      </tr>
20      <tr th:each="book:${books}">
21        <td>[[${book.bookId}]]</td>
22        <td>[[${book.bookName}]]</td>
23        <td>[[${book.publisher}]]</td>
24        <td>[[${book.price}]]</td>
25        <td>
26          <a th:href="@{/book/modify?bookId=${book.bookId}}">수정</a>
27          <a th:href="@{/book/delete?bookId=${book.bookId}}">삭제</a>
28        </td>
29      </tr>
30    </table>
31  </body>
32 </html>

```

### • book list



#### book 목록

[BookStore 메인](#) [book 등록](#)

도서ID	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	<a href="#">수정</a> <a href="#">삭제</a>
2	축구아는 여자	나무수	13000	<a href="#">수정</a> <a href="#">삭제</a>
3	축구의 이해	대한미디어	22000	<a href="#">수정</a> <a href="#">삭제</a>
4	골프 바이블	대한미디어	35000	<a href="#">수정</a> <a href="#">삭제</a>
5	피겨 교본	굿스포츠	8000	<a href="#">수정</a> <a href="#">삭제</a>
6	역도 단계별기술	굿스포츠	6000	<a href="#">수정</a> <a href="#">삭제</a>

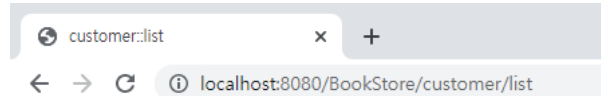
### • customer list

```

1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <meta charset="UTF-8">
5     <title>customer::list</title>
6   </head>
7   <body>
8     <h3>customer 목록</h3>
9     <a href="/BookStore">BookStore 메인</a>
10    <a href="/BookStore/customer/register">customer 등록</a>
11
12    <table border="1">
13      <tr>
14        <th>고객ID</th>
15        <th>고객명</th>
16        <th>주소</th>
17        <th>휴대폰</th>
18        <th>관리</th>
19      </tr>
20      <tr th:each="cust:${custs}">
21        <td>[[${cust.custId}]]</td>
22        <td>[[${cust.name}]]</td>
23        <td>[[${cust.address}]]</td>
24        <td>[[${cust.phone}]]</td>
25        <td>
26          <a th:href="@{/customer/modify?custId=${cust.custId}}">수정</a>
27          <a th:href="@{/customer/delete?custId=${cust.custId}}">삭제</a>
28        </td>
29      </tr>
30    </table>
31  </body>
32 </html>

```

### • customer list



#### customer 목록

[BookStore 메인](#) [customer 등록](#)

고객ID	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스터	000-5000-0001	<a href="#">수정</a> <a href="#">삭제</a>
2	김연아	대한민국 서울	000-6000-0001	<a href="#">수정</a> <a href="#">삭제</a>
3	장미란	대한민국 강원도	000-7000-0001	<a href="#">수정</a> <a href="#">삭제</a>
4	추신수	미국 클리블랜드	000-8000-0001	<a href="#">수정</a> <a href="#">삭제</a>
5	이수근	서울특별시	010-1235-5221	<a href="#">수정</a> <a href="#">삭제</a>

## • book, customer register 코드

### • book register

```
1 <!DOCTYPE html>
2<html xmlns:th="http://www.thymeleaf.org">
3<head>
4    <meta charset="UTF-8">
5    <title>book::register</title>
6</head>
7<body>
8    <h3>book 등록</h3>
9    <a href="/BookStore">BookStore 메인</a>
10   <a href="/BookStore/book/List">book 목록</a>
11
12   <form action="/BookStore/book/register" method="post">
13       <table border="1">
14           <tr>
15               <td>도서명</td>
16               <td><input type="text" name="bookName"></td>
17           </tr>
18           <tr>
19               <td>출판사</td>
20               <td><input type="text" name="publisher"></td>
21           </tr>
22           <tr>
23               <td>가격</td>
24               <td><input type="number" name="price"></td>
25           </tr>
26           <tr>
27               <td colspan="2" align="right"><input type="submit" value="등록"></td>
28           </tr>
29       </table>
30   </form>
31 </body>
32 </html>
```

### • customer register

```
1 <!DOCTYPE html>
2<html xmlns:th="http://www.thymeleaf.org">
3<head>
4    <meta charset="UTF-8">
5    <title>customer::register</title>
6</head>
7<body>
8    <h3>customer 등록</h3>
9    <a href="/BookStore">BookStore 메인</a>
10   <a href="/BookStore/customer/List">customer 목록</a>
11
12   <form action="/BookStore/customer/register" method="post">
13       <table border="1">
14           <tr>
15               <td>고객명</td>
16               <td><input type="text" name="name"></td>
17           </tr>
18           <tr>
19               <td>주소</td>
20               <td><input type="text" name="address"></td>
21           </tr>
22           <tr>
23               <td>휴대폰</td>
24               <td><input type="text" name="phone"></td>
25           </tr>
26           <tr>
27               <td colspan="2" align="right"><input type="submit" value="등록"></td>
28           </tr>
29       </table>
30   </form>
31 </body>
32 </html>
```



## • book, customer modify 코드

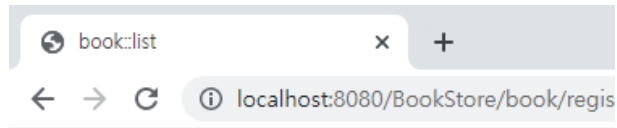
### • book modify

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <meta charset="UTF-8">
5     <title>book::modify</title>
6   </head>
7   <body>
8     <h3>book 수정</h3>
9     <a href="/BookStore">BookStore 메인</a>
10    <a href="/BookStore/book/List">book 목록</a>
11
12    <form th:action="@{/book/modify?(bookId=${book.bookId})}" method="post">
13      <table border="1">
14        <tr>
15          <td>도서ID</td>
16          <td><input type="text" name="bookId" th:value="${book.bookId}" readonly="readonly"></td>
17        </tr>
18        <tr>
19          <td>도서명</td>
20          <td><input type="text" name="bookName" th:value="${book.bookName}"></td>
21        </tr>
22        <tr>
23          <td>출판사</td>
24          <td><input type="text" name="publisher" th:value="${book.publisher}"></td>
25        </tr>
26        <tr>
27          <td>가격</td>
28          <td><input type="number" name="price" th:value="${book.price}"></td>
29        </tr>
30        <tr>
31          <td colspan="2" align="right"><input type="submit" value="등록"></td>
32        </tr>
33      </table>
34    </form>
35  </body>
36 </html>
```

### • customer modify

```
1 <!DOCTYPE html>
2 <html xmlns:th="http://www.thymeleaf.org">
3   <head>
4     <meta charset="UTF-8">
5     <title>customer::modify</title>
6   </head>
7   <body>
8     <h3>customer 수정</h3>
9     <a href="/BookStore">BookStore 메인</a>
10    <a href="/BookStore/customer/List">customer 목록</a>
11
12    <form th:action="@{/customer/modify?(custId=${cust.custId})}" method="post">
13      <table border="1">
14        <tr>
15          <td>고객ID</td>
16          <td><input type="text" name="custId" th:value="${cust.custId}" readonly="readonly"></td>
17        </tr>
18        <tr>
19          <td>고객명</td>
20          <td><input type="text" name="name" th:value="${cust.name}"></td>
21        </tr>
22        <tr>
23          <td>주소</td>
24          <td><input type="text" name="address" th:value="${cust.address}"></td>
25        </tr>
26        <tr>
27          <td>휴대폰</td>
28          <td><input type="text" name="phone" th:value="${cust.phone}"></td>
29        </tr>
30        <tr>
31          <td colspan="2" align="right"><input type="submit" value="등록"></td>
32        </tr>
33      </table>
34    </form>
35  </body>
36 </html>
```

## • book register

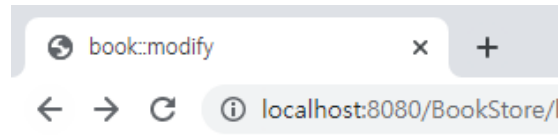


### book 등록

[BookStore 메인 book 목록](#)

도서명	다원상 수상자들
출판사	김다원
가격	82000
<input type="button" value="등록"/>	

## • book modify

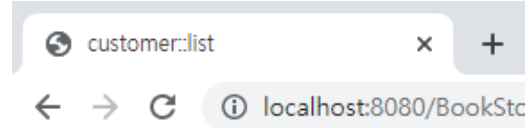


### book 수정

[BookStore 메인 book 목록](#)

도서ID	24
도서명	노벨상 수상자들
출판사	김노벨
가격	182000
<input type="button" value="등록"/>	

## • customer register

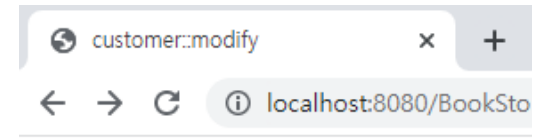


### customer 등록

[BookStore 메인 customer 목록](#)

고객명	강호동
주소	경상남도 마산
휴대폰	010-1234-5656
<input type="button" value="등록"/>	

## • customer modify



### customer 수정

[BookStore 메인 customer 목록](#)

고객ID	11
고객명	송민호
주소	서울특별시
휴대폰	010-2424-4242
<input type="button" value="등록"/>	

### 3. 기능 구현

- book(도서) 기본 CRUD

- customer(고객) 기본 CRUD

- Mybatis 사용

- book, customer mappers xml

- book

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE mapper
3   PUBLIC "-//mybatis.org/DTD Mapper 3.0//EN"
4   "https://mybatis.org/dtd/mybatis-3-mapper.dtd">
5 <mapper namespace="book">
6
7   <insert id="insertBook">
8     INSERT INTO `book` (bookName, publisher, price) VALUES ({bookName}, #{publisher}, #{price});
9   </insert>
10
11   <select id="selectBook" resultType="kr.co.bookstore.vo.BookVO">
12     select * from `book` where `bookId` = #{bookId};
13   </select>
14
15   <select id="selectBooks" resultType="kr.co.bookstore.vo.BookVO">
16     select * from `book`;
17   </select>
18
19   <update id="updateBook">
20     update `book` set
21       `bookName`=#{bookName},
22       `publisher`=#{publisher},
23       `price`=#{price}
24     where
25       `bookId`=#{bookId};
26   </update>
27   <delete id="deleteBook">
28     delete from `book` where `bookId`=#{bookId};
29   </delete>
30 </mapper>
```

- customer

```
-//mybatis.org/DTD Mapper 3.0//EN (doctype)
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE mapper
3   PUBLIC "-//mybatis.org/DTD Mapper 3.0//EN"
4   "https://mybatis.org/dtd/mybatis-3-mapper.dtd">
5 <mapper namespace="cust">
6
7   <insert id="insertCust">
8     INSERT INTO `customer` (name, address, phone) VALUES ({name}, #{address}, #{phone});
9   </insert>
10
11   <select id="selectCust" resultType="kr.co.bookstore.vo.CustVO">
12     select * from `customer` where `custId` = #{custId};
13   </select>
14
15   <select id="selectCusts" resultType="kr.co.bookstore.vo.CustVO">
16     select * from `customer`;
17   </select>
18
19   <update id="updateCust">
20     update `customer` set
21       `name`=#{name},
22       `address`=#{address},
23       `phone`=#{phone}
24     where
25       `custId`=#{custId};
26   </update>
27   <delete id="deleteCust">
28     delete from `customer` where `custId`=#{custId};
29   </delete>
30 </mapper>
```

## • book

```
1 package kr.co.bookstore.vo;
2
3 *import lombok.AllArgsConstructor;
4
5 @Getter
6 @Setter
7 @AllArgsConstructor
8 @NoArgsConstructor
9 @ToString
10 public class BookVO {
11     private int bookId;
12     private String bookName;
13     private String publisher;
14     private int price;
15 }
16
```

## • VO

```
1 package kr.co.bookstore.dao;
2
3 *import java.util.List;
4
5 @Repository
6 public class BookDAO {
7
8     @Autowired
9     private SqlSessionTemplate mybatis;
10
11     public void insertBook(BookVO vo) {
12         mybatis.insert("book.insertBook", vo);
13     }
14
15     public BookVO selectBook(int bookId) {
16         return mybatis.selectOne("book.selectBook", bookId);
17     }
18
19     public List<BookVO> selectBooks() {
20         return mybatis.selectList("book.selectBooks");
21     }
22
23     public void updateBook(BookVO vo) {
24         mybatis.update("book.updateBook", vo);
25     }
26
27     public void deleteBook(int bookId) {
28         mybatis.delete("book.deleteBook", bookId);
29     }
30 }
31
```

## • DAO

```
1 package kr.co.bookstore.service;
2
3 *import java.util.List;
4
5 @Service
6 public class BookService {
7
8     @Autowired
9     private BookDAO dao;
10
11     public void insertBook(BookVO vo) {
12         dao.insertBook(vo);
13     }
14
15     public BookVO selectBook(int bookId) {
16         return dao.selectBook(bookId);
17     }
18
19     public List<BookVO> selectBooks() {
20         return dao.selectBooks();
21     }
22
23     public void updateBook(BookVO vo) {
24         dao.updateBook(vo);
25     }
26
27     public void deleteBook(int bookId) {
28         dao.deleteBook(bookId);
29     }
30 }
31
```

## • Service

```
15 @Controller
16 public class BookController {
17
18     @Autowired
19     private BookService service;
20
21     @GetMapping("/book/list")
22     public String list(Model model) {
23         List<BookVO> books = service.selectBooks();
24         model.addAttribute("books", books);
25         return "/book/list";
26     }
27
28     @GetMapping("/book/register")
29     public String register() {
30         return "/book/register";
31     }
32
33     @PostMapping("/book/register")
34     public String register(BookVO vo) {
35         service.insertBook(vo);
36         return "redirect:/book/list";
37     }
38
39     @GetMapping("/book/modify")
40     public String modify(int bookId, Model model) {
41         BookVO book = service.selectBook(bookId);
42         model.addAttribute("book", book);
43         return "/book/modify";
44     }
45
46     @PostMapping("/book/modify")
47     public String modify(BookVO vo) {
48         service.updateBook(vo);
49         return "redirect:/book/list";
50     }
51
52     @GetMapping("/book/delete")
53     public String delete(int bookId) {
54         service.deleteBook(bookId);
55         return "redirect:/book/list";
56     }
57 }
58
```

## • Controller

## • customer

## • VO

```
1 package kr.co.bookstore.vo;
2
3 import lombok.AllArgsConstructor;
4
5 @Getter
6 @Setter
7 @AllArgsConstructor
8 @NoArgsConstructor
9 @ToString
10 public class CustVO {
11     private int custId;
12     private String name;
13     private String address;
14     private String phone;
15 }
16
```

## • DAO

```
1 package kr.co.bookstore.dao;
2
3 import java.util.List;
4
5 import org.mybatis.spring.SqlSessionTemplate;
6 import org.springframework.beans.factory.annotation.Autowired;
7 import org.springframework.stereotype.Repository;
8
9 import kr.co.bookstore.vo.CustVO;
10
11 @Repository
12 public class CustDAO {
13
14     @Autowired
15     private SqlSessionTemplate mybatis;
16
17     public void insertCust(CustVO vo) {
18         mybatis.insert("cust.insertCust", vo);
19     }
20
21     public CustVO selectCust(int custId) {
22         return mybatis.selectOne("cust.selectCust", custId);
23     }
24
25     public List<CustVO> selectCusts() {
26         return mybatis.selectList("cust.selectCusts");
27     }
28
29     public void updateCust(CustVO vo) {
30         mybatis.update("cust.updateCust", vo);
31     }
32
33     public void deleteCust(int custId) {
34         mybatis.delete("cust.deleteCust", custId);
35     }
36 }
```

## • Service

```
1 package kr.co.bookstore.service;
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Service;
7
8 import kr.co.bookstore.dao.CustDAO;
9 import kr.co.bookstore.vo.CustVO;
10
11 @Service
12 public class CustService {
13
14     @Autowired
15     private CustDAO dao;
16
17     public void insertCust(CustVO vo) {
18         dao.insertCust(vo);
19     }
20
21     public CustVO selectCust(int custId) {
22         return dao.selectCust(custId);
23     }
24
25     public List<CustVO> selectCusts() {
26         return dao.selectCusts();
27     }
28
29     public void updateCust(CustVO vo) {
30         dao.updateCust(vo);
31     }
32
33     public void deleteCust(int custId) {
34         dao.deleteCust(custId);
35     }
36 }
```

## • Controller

```
15 @Controller
16 public class CustController {
17
18     @Autowired
19     private CustService service;
20
21     @GetMapping("/customer/list")
22     public String list(Model model) {
23         List<CustVO> custs = service.selectCusts();
24         model.addAttribute("custs", custs);
25         return "/customer/list";
26     }
27
28     @GetMapping("/customer/register")
29     public String register() {
30         return "/customer/register";
31     }
32
33     @PostMapping("/customer/register")
34     public String register(CustVO vo) {
35         service.insertCust(vo);
36         return "redirect:/customer/list";
37     }
38
39     @GetMapping("/customer/modify")
40     public String modify(int custId, Model model) {
41         CustVO cust = service.selectCust(custId);
42         model.addAttribute("cust", cust);
43         return "/customer/modify";
44     }
45
46     @PostMapping("/customer/modify")
47     public String modify(CustVO vo) {
48         service.updateCust(vo);
49         return "redirect:/customer/list";
50     }
51
52     @GetMapping("/customer/delete")
53     public String delete(int custId) {
54         service.deleteCust(custId);
55         return "redirect:/customer/list";
56     }
57 }
```

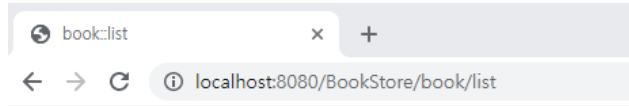
## 4. 실행

- book 등록, 수정, 삭제 기능  
동작 화면 첨부
- customer  
등록, 수정, 삭제 기능  
동작 화면 첨부

※ 목록, 등록, 수정 화면은 이전 단계에서 첨부

## • book

## • 등록



### book 목록

BookStore 메인 book 등록

도서ID	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	<a href="#">수정</a> <a href="#">삭제</a>
2	축구하는 여자	나무수	13000	<a href="#">수정</a> <a href="#">삭제</a>
3	축구의 이해	대한미디어	22000	<a href="#">수정</a> <a href="#">삭제</a>
4	골프 바이블	대한미디어	35000	<a href="#">수정</a> <a href="#">삭제</a>
5	피겨 교본	굿스포츠	8000	<a href="#">수정</a> <a href="#">삭제</a>
6	역도 단계별기술	굿스포츠	6000	<a href="#">수정</a> <a href="#">삭제</a>
24	다윈상 수상자들	김다원	82000	<a href="#">수정</a> <a href="#">삭제</a>
25	정보처리기사 필기+실기	공무원	42000	<a href="#">수정</a> <a href="#">삭제</a>
26	나는 순하게 살기로 했다	진라면	19000	<a href="#">수정</a> <a href="#">삭제</a>

HeidiSQL 12.1.0.6537

도움말

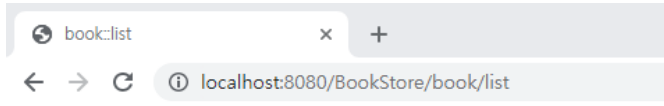
호스트: 127.0.0.1    데이터베이스: java2\_bookstore

테이블: book    데이터

java2\_bookstore.book: 9 >> 다음    모두 보기    정렬    열 (4/4)

bookId	bookName	publisher	price
1	축구의 역사	굿스포츠	7,000
2	축구하는 여자	나무수	13,000
3	축구의 이해	대한미디어	22,000
4	골프 바이블	대한미디어	35,000
5	피겨 교본	굿스포츠	8,000
6	역도 단계별기술	굿스포츠	6,000
24	다윈상 수상자들	김다원	82,000
25	정보처리기사 필기+실기	공무원	42,000
26	나는 순하게 살기로 했다	진라면	19,000

## • 수정



### book 목록

BookStore 메인 book 등록

도서ID	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	<a href="#">수정</a> <a href="#">삭제</a>
2	축구하는 여자	나무수	13000	<a href="#">수정</a> <a href="#">삭제</a>
3	축구의 이해	대한미디어	22000	<a href="#">수정</a> <a href="#">삭제</a>
4	골프 바이블	대한미디어	35000	<a href="#">수정</a> <a href="#">삭제</a>
5	피겨 교본	굿스포츠	8000	<a href="#">수정</a> <a href="#">삭제</a>
6	역도 단계별기술	굿스포츠	6000	<a href="#">수정</a> <a href="#">삭제</a>
24	노벨상 수상자들	김노벨	182000	<a href="#">수정</a> <a href="#">삭제</a>
25	정보처리기사 필기+실기	공무원	42000	<a href="#">수정</a> <a href="#">삭제</a>
26	나는 순하게 살기로 했다	진라면	19000	<a href="#">수정</a> <a href="#">삭제</a>

HeidiSQL 12.1.0.6537

도움말

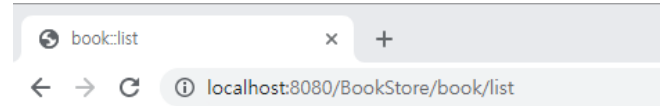
호스트: 127.0.0.1    데이터베이스: java2\_bookstore

테이블: book    데이터

java2\_bookstore.book: 9 >> 다음    모두 보기    정렬    열 (4/4)

bookId	bookName	publisher	price
1	축구의 역사	굿스포츠	7,000
2	축구하는 여자	나무수	13,000
3	축구의 이해	대한미디어	22,000
4	골프 바이블	대한미디어	35,000
5	피겨 교본	굿스포츠	8,000
6	역도 단계별기술	굿스포츠	6,000
24	노벨상 수상자들	김노벨	182,000
25	정보처리기사 필기+실기	공무원	42,000
26	나는 순하게 살기로 했다	진라면	19,000

## • 삭제



### book 목록

BookStore 메인 book 등록

도서ID	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	<a href="#">수정</a> <a href="#">삭제</a>
2	축구하는 여자	나무수	13000	<a href="#">수정</a> <a href="#">삭제</a>
3	축구의 이해	대한미디어	22000	<a href="#">수정</a> <a href="#">삭제</a>
4	골프 바이블	대한미디어	35000	<a href="#">수정</a> <a href="#">삭제</a>
5	피겨 교본	굿스포츠	8000	<a href="#">수정</a> <a href="#">삭제</a>
6	역도 단계별기술	굿스포츠	6000	<a href="#">수정</a> <a href="#">삭제</a>
25	정보처리기사 필기+실기	공무원	42000	<a href="#">수정</a> <a href="#">삭제</a>
26	나는 순하게 살기로 했다	진라면	19000	<a href="#">수정</a> <a href="#">삭제</a>

HeidiSQL 12.1.0.6537

도움말

호스트: 127.0.0.1    데이터베이스: java2\_bookstore

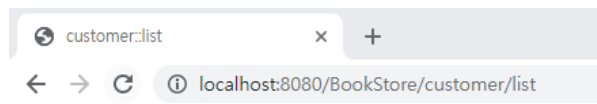
테이블: book    데이터

java2\_bookstore.book: 8 >> 다음    모두 보기    정렬    열 (4/4)

bookId	bookName	publisher	price
1	축구의 역사	굿스포츠	7,000
2	축구하는 여자	나무수	13,000
3	축구의 이해	대한미디어	22,000
4	골프 바이블	대한미디어	35,000
5	피겨 교본	굿스포츠	8,000
6	역도 단계별기술	굿스포츠	6,000
25	정보처리기사 필기+실기	공무원	42,000
26	나는 순하게 살기로 했다	진라면	19,000

## • customer

### • 등록



#### customer 목록

[BookStore 메인 customer 등록](#)

고객ID	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스타	000-5000-0001	<a href="#">수정</a> <a href="#">삭제</a>
2	김연아	대한민국 서울	000-6000-0001	<a href="#">수정</a> <a href="#">삭제</a>
3	장미란	대한민국 강원도	000-7000-0001	<a href="#">수정</a> <a href="#">삭제</a>
4	추신수	미국 클리블랜드	000-8000-0001	<a href="#">수정</a> <a href="#">삭제</a>
5	이수근	서울특별시	010-1235-5221	<a href="#">수정</a> <a href="#">삭제</a>
10	이승기	서울특별시	010-5252-8282	<a href="#">수정</a> <a href="#">삭제</a>
11	강호동	경상남도 마산	010-2424-4242	<a href="#">수정</a> <a href="#">삭제</a>
12	은지원	서울특별시	010-4646-7979	<a href="#">수정</a> <a href="#">삭제</a>

f - HeidiSQL 12.1.0.6537

도움말

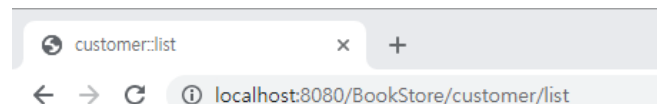
호스트: 127.0.0.1    데이터베이스: java2\_bookstore

테이블: customer    데이터

java2\_bookstore.customer >> 다음    모두 보기    정렬    열 (4)

custid	name	address	phone
1	박지성	영국 맨체스타	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
3	장미란	대한민국 강원도	000-7000-0001
4	추신수	미국 클리블랜드	000-8000-0001
5	이수근	서울특별시	010-1235-5221
10	이승기	서울특별시	010-5252-8282
11	강호동	경상남도 마산	010-2424-4242
12	은지원	서울특별시	010-4646-7979

### • 수정



#### customer 목록

[BookStore 메인 customer 등록](#)

고객ID	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스타	000-5000-0001	<a href="#">수정</a> <a href="#">삭제</a>
2	김연아	대한민국 서울	000-6000-0001	<a href="#">수정</a> <a href="#">삭제</a>
3	장미란	대한민국 강원도	000-7000-0001	<a href="#">수정</a> <a href="#">삭제</a>
4	추신수	미국 클리블랜드	000-8000-0001	<a href="#">수정</a> <a href="#">삭제</a>
5	이수근	서울특별시	010-1235-5221	<a href="#">수정</a> <a href="#">삭제</a>
10	이승기	서울특별시	010-5252-8282	<a href="#">수정</a> <a href="#">삭제</a>
11	송민호	서울특별시	010-2424-4242	<a href="#">수정</a> <a href="#">삭제</a>
12	은지원	서울특별시	010-4646-7979	<a href="#">수정</a> <a href="#">삭제</a>

f - HeidiSQL 12.1.0.6537

도움말

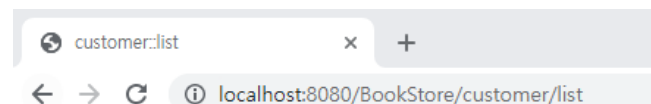
호스트: 127.0.0.1    데이터베이스: java2\_bookstore

테이블: customer    데이터

java2\_bookstore.customer >> 다음    모두 보기    정렬    열 (4/4)

custid	name	address	phone
1	박지성	영국 맨체스타	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
3	장미란	대한민국 강원도	000-7000-0001
4	추신수	미국 클리블랜드	000-8000-0001
5	이수근	서울특별시	010-1235-5221
10	이승기	서울특별시	010-5252-8282
11	송민호	서울특별시	010-2424-4242
12	은지원	서울특별시	010-4646-7979

### • 삭제



#### customer 목록

[BookStore 메인 customer 등록](#)

고객ID	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스타	000-5000-0001	<a href="#">수정</a> <a href="#">삭제</a>
2	김연아	대한민국 서울	000-6000-0001	<a href="#">수정</a> <a href="#">삭제</a>
3	장미란	대한민국 강원도	000-7000-0001	<a href="#">수정</a> <a href="#">삭제</a>
4	추신수	미국 클리블랜드	000-8000-0001	<a href="#">수정</a> <a href="#">삭제</a>
5	이수근	서울특별시	010-1235-5221	<a href="#">수정</a> <a href="#">삭제</a>
10	이승기	서울특별시	010-5252-8282	<a href="#">수정</a> <a href="#">삭제</a>
12	은지원	서울특별시	010-4646-7979	<a href="#">수정</a> <a href="#">삭제</a>

f - HeidiSQL 12.1.0.6537

도움말

호스트: 127.0.0.1    데이터베이스: java2\_bookstore

테이블: customer    데이터

java2\_bookstore.customer >> 다음    모두 보기    정렬    열 (4)

custid	name	address	phone
1	박지성	영국 맨체스타	000-5000-0001
2	김연아	대한민국 서울	000-6000-0001
3	장미란	대한민국 강원도	000-7000-0001
4	추신수	미국 클리블랜드	000-8000-0001
5	이수근	서울특별시	010-1235-5221
10	이승기	서울특별시	010-5252-8282
12	은지원	서울특별시	010-4646-7979