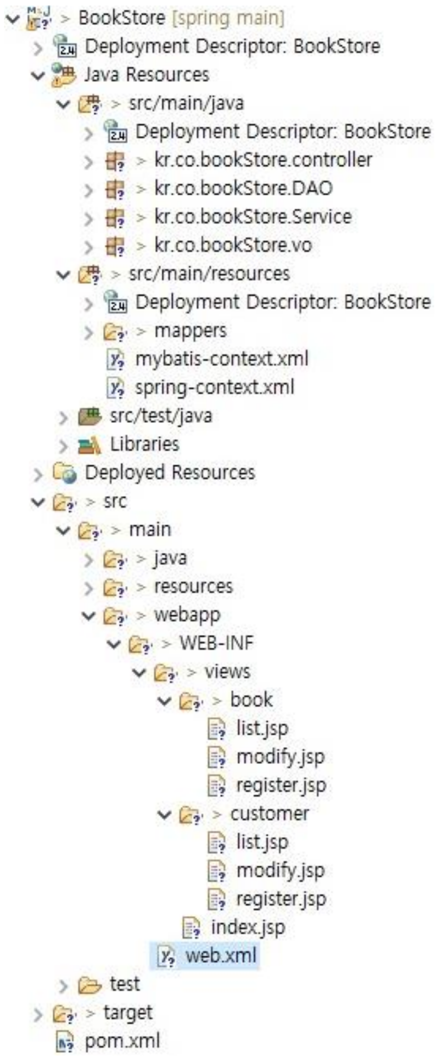
모던 웹 애플리케이션 개발2

2023.01.04 박진휘

Part 1 프로젝트 생성 및 구성



```
BookStore [spring main]
  Deployment Descriptor: BookStore
  Java Resources
    src/main/java
      Deployment Descriptor: BookStore
      kr.co.bookStore.controller
      kr.co.bookStore.DAO
      kr.co.bookStore.Service
      kr.co.bookStore.vo
    src/main/resources
      Deployment Descriptor: BookStore
      mappers
      mybatis-context.xml
      spring-context.xml
    src/test/java
    Libraries
  Deployed Resources
  src
    main
      java
      resources
      webapp
        WEB-INF
          views
            book
              list.jsp
              modify.jsp
              register.jsp
            customer
              list.jsp
              modify.jsp
              register.jsp
            index.jsp
          web.xml
    test
  target
  pom.xml
```

ProjectName : BookStore

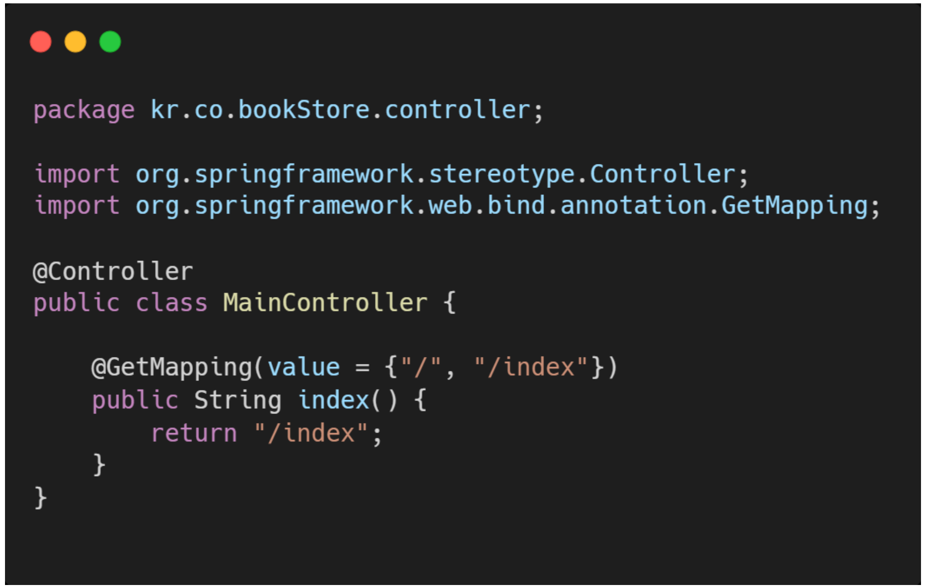
WAS : Tomcat 9

DB : java1_bookstore

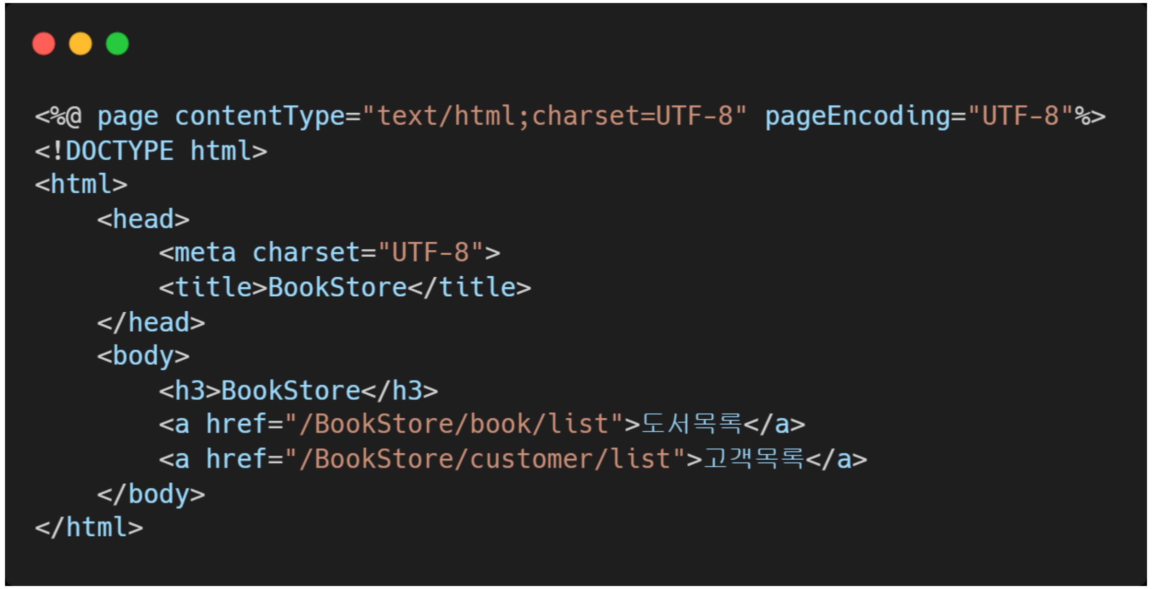
Framework : Spring, Mybatis

Tools : Eclipse, Workbench

Part 2 화면구현



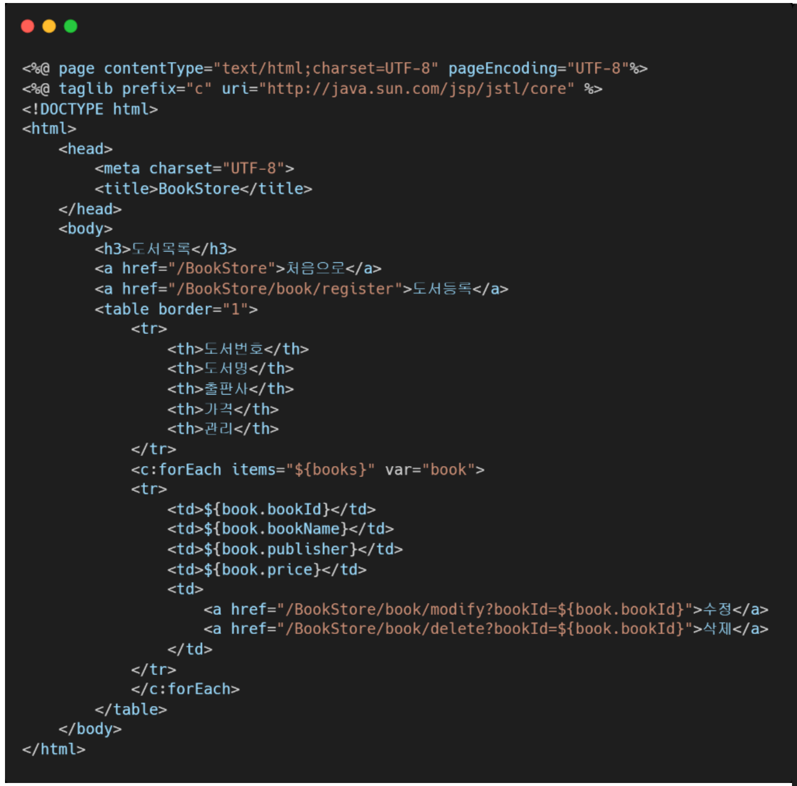
```java
package kr.co.bookStore.controller;

import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.GetMapping;

@Controller
public class MainController {

    @GetMapping(value = {"/", "/index"})
    public String index() {
        return "/index";
    }
}
```



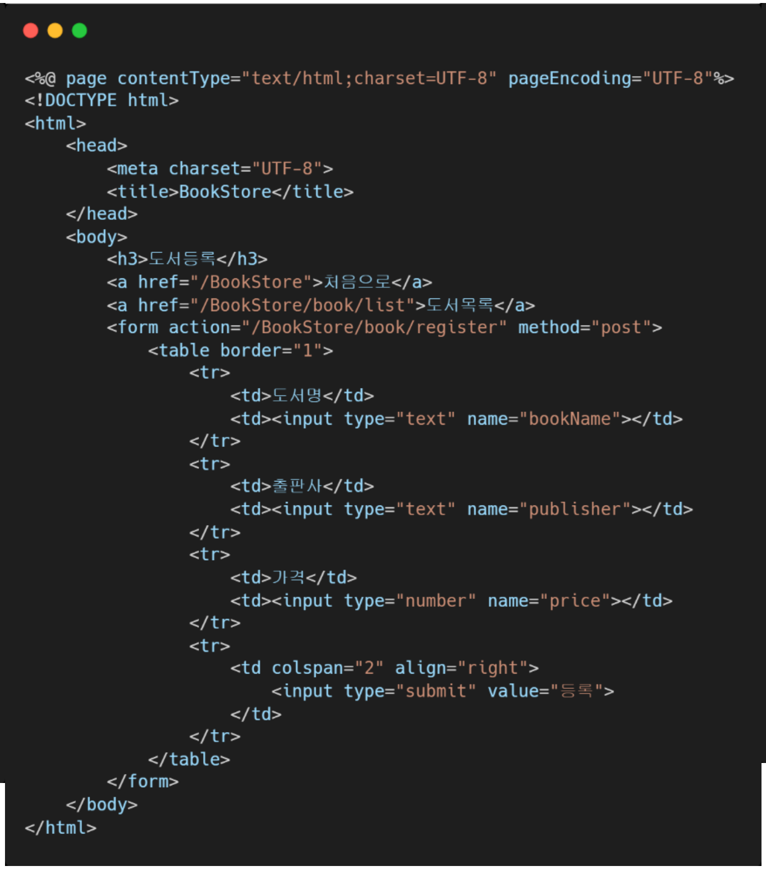
```jsp
<%@ page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>BookStore</title>
    </head>
    <body>
        <h3>BookStore</h3>
        <a href="/BookStore/book/list">도서목록</a>
        <a href="/BookStore/customer/list">고객목록</a>
    </body>
</html>
```

IndexController, Index.jsp

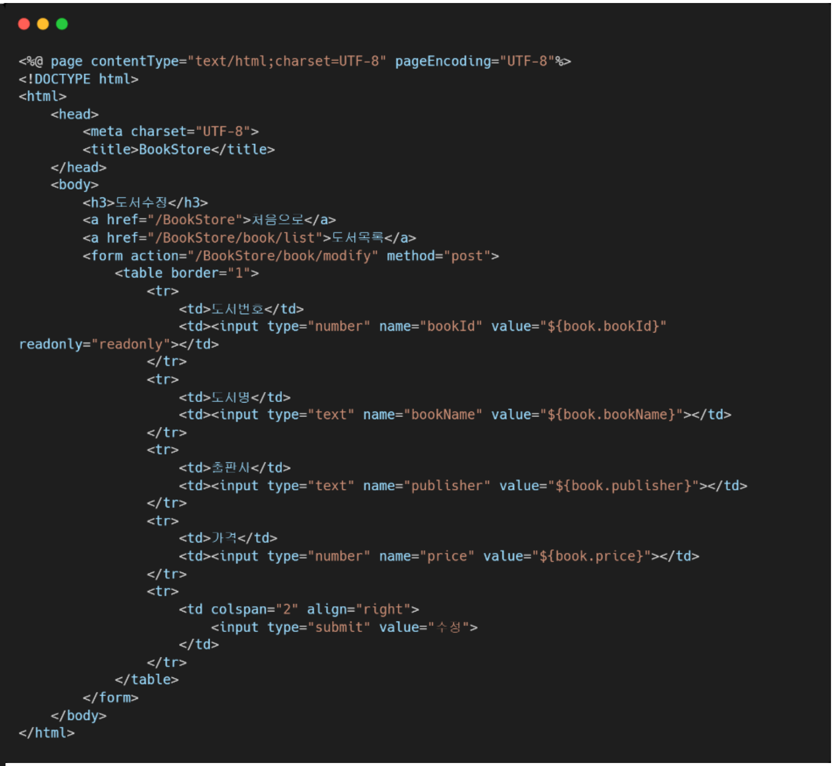
Part 2 화면구현



```
<%@ page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>BookStore</title>
    </head>
    <body>
        <h3>도서목록</h3>
        <a href="/BookStore">처음으로</a>
        <a href="/BookStore/book/register">도서등록</a>
        <table border="1">
            <tr>
                <th>도서번호</th>
                <th>도서명</th>
                <th>출판사</th>
                <th>가격</th>
                <th>관리</th>
            </tr>
            <c:forEach items="${books}" var="book">
            <tr>
                <td>${book.bookId}</td>
                <td>${book.bookName}</td>
                <td>${book.publisher}</td>
                <td>${book.price}</td>
                <td>
                    <a href="/BookStore/book/modify?bookId=${book.bookId}">수정</a>
                    <a href="/BookStore/book/delete?bookId=${book.bookId}">삭제</a>
                </td>
            </tr>
            </c:forEach>
        </table>
    </body>
</html>
```



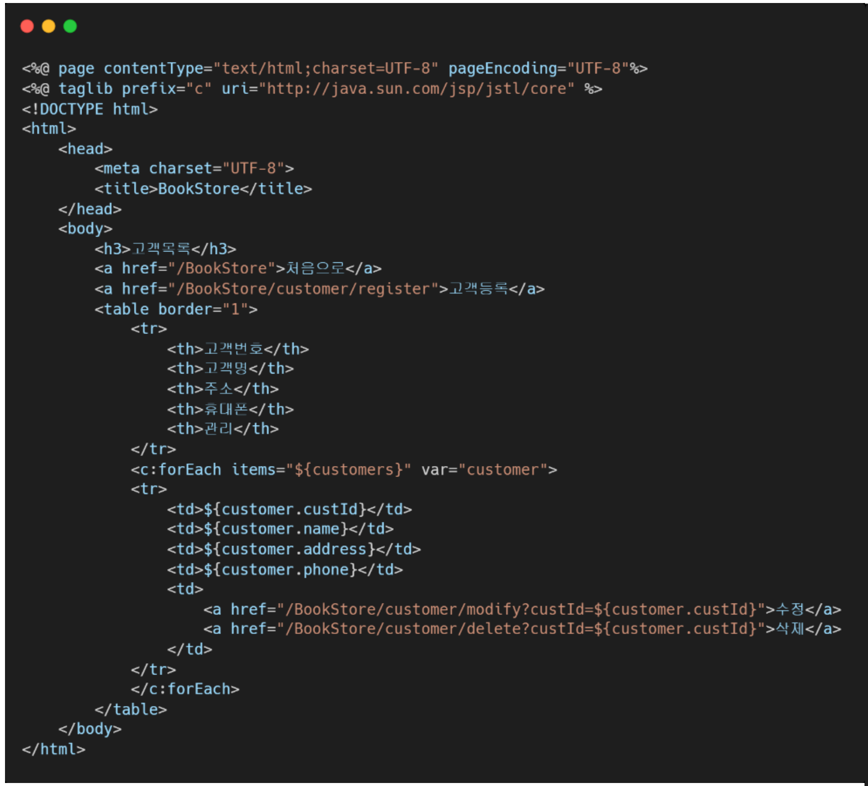
```
<%@ page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>BookStore</title>
    </head>
    <body>
        <h3>도서등록</h3>
        <a href="/BookStore">처음으로</a>
        <a href="/BookStore/book/list">도서목록</a>
        <form action="/BookStore/book/register" method="post">
            <table border="1">
                <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="number" name="price"></td>
                </tr>
                <tr>
                    <td colspan="2" align="right">
                        <input type="submit" value="등록">
                    </td>
                </tr>
            </table>
        </form>
    </body>
</html>
```



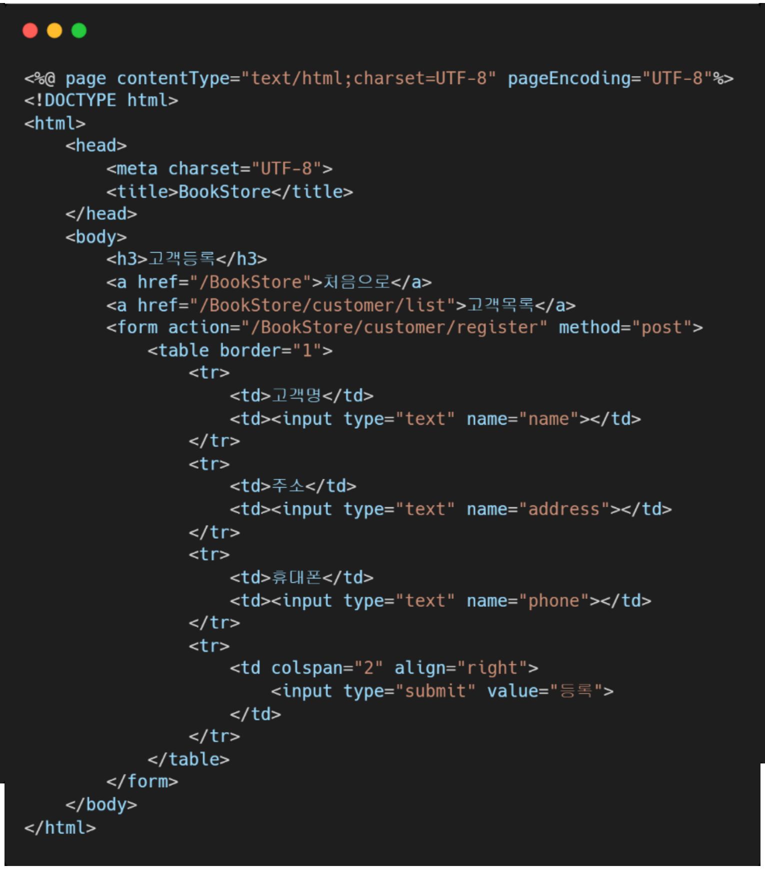
```
<%@ page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>BookStore</title>
    </head>
    <body>
        <h3>도서수정</h3>
        <a href="/BookStore">처음으로</a>
        <a href="/BookStore/book/list">도서목록</a>
        <form action="/BookStore/book/modify" method="post">
            <table border="1">
                <tr>
                    <td>도서번호</td>
                    <td><input type="number" name="bookId" value="${book.bookId}"
readonly="readonly"></td>
                </tr>
                <tr>
                    <td>도서명</td>
                    <td><input type="text" name="bookName" value="${book.bookName}"></td>
                </tr>
                <tr>
                    <td>출판사</td>
                    <td><input type="text" name="publisher" value="${book.publisher}"></td>
                </tr>
                <tr>
                    <td>가격</td>
                    <td><input type="number" name="price" value="${book.price}"></td>
                </tr>
                <tr>
                    <td colspan="2" align="right">
                        <input type="submit" value="수정">
                    </td>
                </tr>
            </table>
        </form>
    </body>
</html>
```

Book list.jsp, register.jsp, modify.jsp

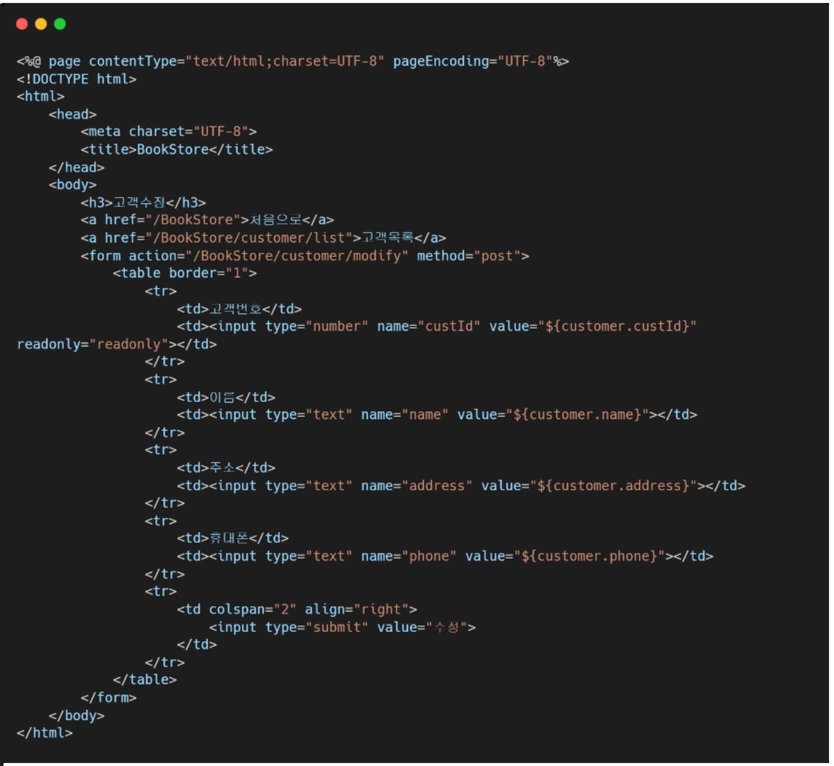
Part 2 화면구현



```
<%@ page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>BookStore</title>
    </head>
    <body>
        <h3>고객목록</h3>
        <a href="/BookStore">처음으로</a>
        <a href="/BookStore/customer/register">고객등록</a>
        <table border="1">
            <tr>
                <th>고객번호</th>
                <th>고객명</th>
                <th>주소</th>
                <th>휴대폰</th>
                <th>관리</th>
            </tr>
            <c:forEach items="${customers}" var="customer">
            <tr>
                <td>${customer.custId}</td>
                <td>${customer.name}</td>
                <td>${customer.address}</td>
                <td>${customer.phone}</td>
                <td>
                    <a href="/BookStore/customer/modify?custId=${customer.custId}">수정</a>
                    <a href="/BookStore/customer/delete?custId=${customer.custId}">삭제</a>
                </td>
            </tr>
            </c:forEach>
        </table>
    </body>
</html>
```

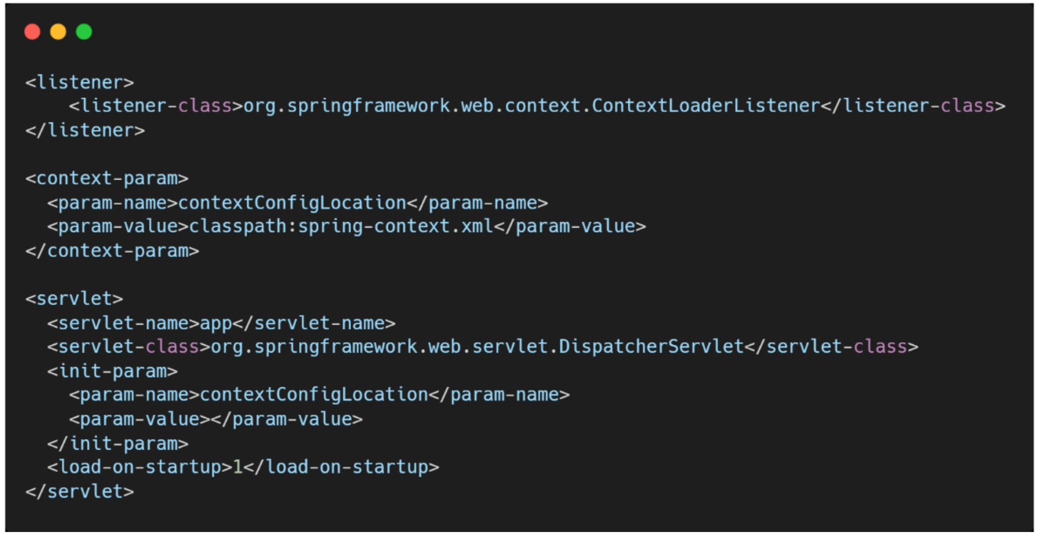


```
<%@ page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>BookStore</title>
    </head>
    <body>
        <h3>고객등록</h3>
        <a href="/BookStore">처음으로</a>
        <a href="/BookStore/customer/list">고객목록</a>
        <form action="/BookStore/customer/register" 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>
    </body>
</html>
```

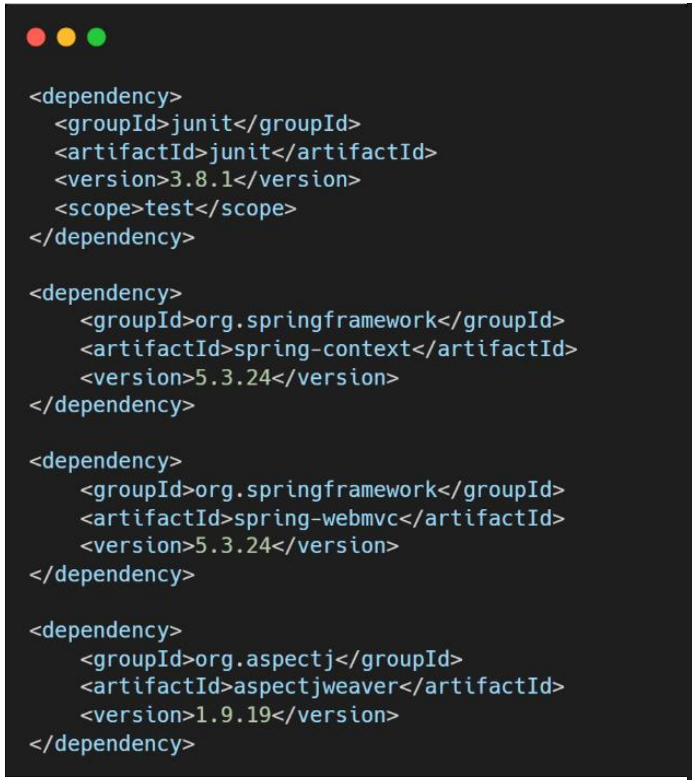


```
<%@ page contentType="text/html;charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title>BookStore</title>
    </head>
    <body>
        <h3>고객수정</h3>
        <a href="/BookStore">처음으로</a>
        <a href="/BookStore/customer/list">고객목록</a>
        <form action="/BookStore/customer/modify" method="post">
            <table border="1">
                <tr>
                    <td>고객번호</td>
                    <td><input type="number" name="custId" value="${customer.custId}"
readonly="readonly"></td>
                </tr>
                <tr>
                    <td>이름</td>
                    <td><input type="text" name="name" value="${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 colspan="2" align="right">
                        <input type="submit" value="수성">
                    </td>
                </tr>
            </table>
        </form>
    </body>
</html>
```

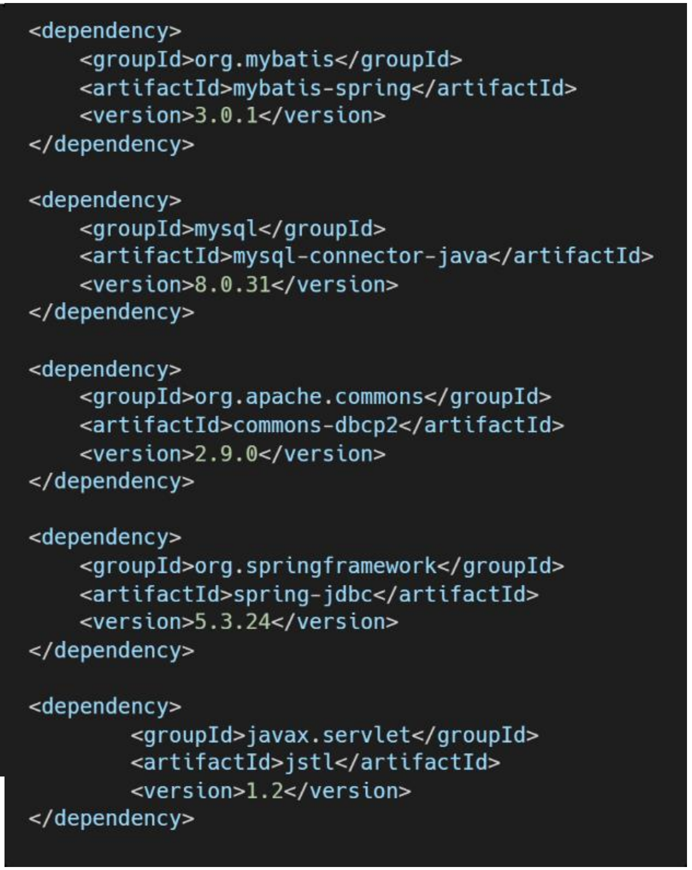
Customer list.jsp, register.jsp, modify.jsp



```
<listener>
    <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>

<context-param>
  <param-name>contextConfigLocation</param-name>
  <param-value>classpath:spring-context.xml</param-value>
</context-param>

<servlet>
  <servlet-name>app</servlet-name>
  <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
  <init-param>
    <param-name>contextConfigLocation</param-name>
    <param-value></param-value>
  </init-param>
  <load-on-startup>1</load-on-startup>
</servlet>
```



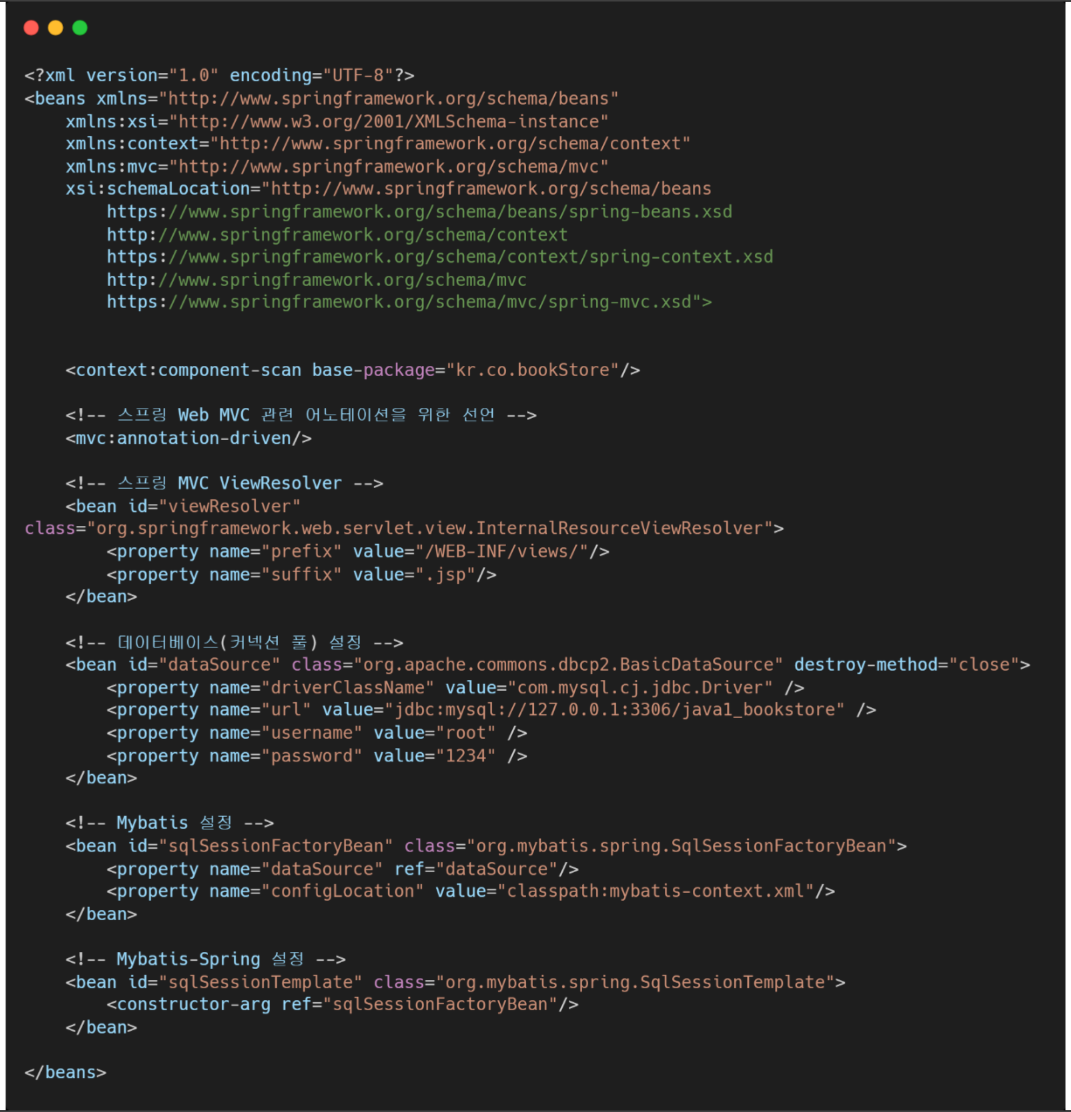
```
<dependency>
  <groupId>junit</groupId>
  <artifactId>junit</artifactId>
  <version>3.8.1</version>
  <scope>test</scope>
</dependency>

<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-context</artifactId>
    <version>5.3.24</version>
</dependency>

<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-webmvc</artifactId>
    <version>5.3.24</version>
</dependency>

<dependency>
    <groupId>org.aspectj</groupId>
    <artifactId>aspectjweaver</artifactId>
    <version>1.9.19</version>
</dependency>
```



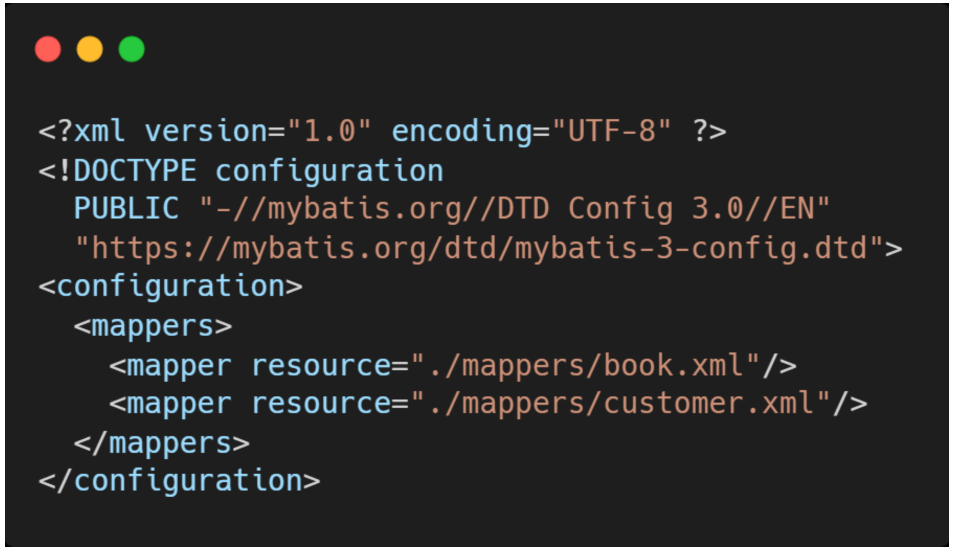
```
<dependency>
    <groupId>org.mybatis</groupId>
    <artifactId>mybatis-spring</artifactId>
    <version>3.0.1</version>
</dependency>

<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.31</version>
</dependency>

<dependency>
    <groupId>org.apache.commons</groupId>
    <artifactId>commons-dbcp2</artifactId>
    <version>2.9.0</version>
</dependency>

<dependency>
    <groupId>org.springframework</groupId>
    <artifactId>spring-jdbc</artifactId>
    <version>5.3.24</version>
</dependency>

<dependency>
        <groupId>javax.servlet</groupId>
        <artifactId>jstl</artifactId>
        <version>1.2</version>
</dependency>
```

web.xml, Maven 설정

Part 3 기능구현



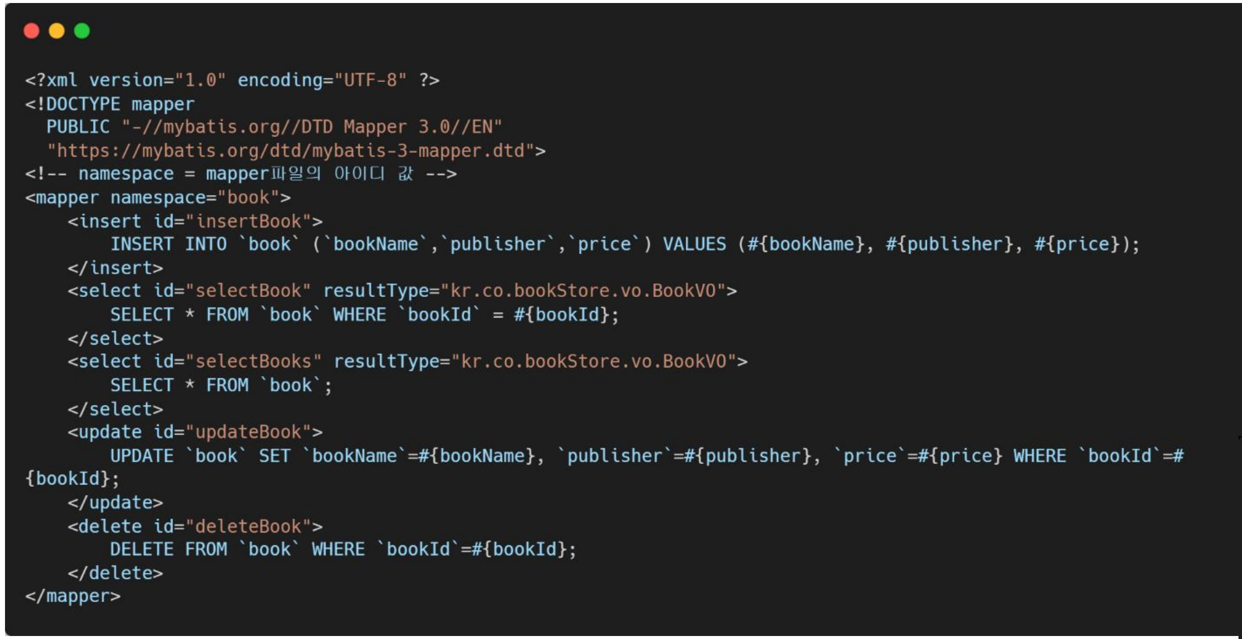
```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xmlns:mvc="http://www.springframework.org/schema/mvc"
    xsi:schemaLocation="http://www.springframework.org/schema/beans
        https://www.springframework.org/schema/beans/spring-beans.xsd
        http://www.springframework.org/schema/context
        https://www.springframework.org/schema/context/spring-context.xsd
        http://www.springframework.org/schema/mvc
        https://www.springframework.org/schema/mvc/spring-mvc.xsd">


    <context:component-scan base-package="kr.co.bookStore"/>

    <!-- 스프링 Web MVC 관련 어노테이션을 위한 선언 -->
    <mvc:annotation-driven/>

    <!-- 스프링 MVC ViewResolver -->
    <bean id="viewResolver"
class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <property name="prefix" value="/WEB-INF/views/"/>
        <property name="suffix" value=".jsp"/>
    </bean>

    <!-- 데이터베이스(커넥션 풀) 설정 -->
    <bean id="dataSource" class="org.apache.commons.dbcp2.BasicDataSource" destroy-method="close">
        <property name="driverClassName" value="com.mysql.cj.jdbc.Driver" />
        <property name="url" value="jdbc:mysql://127.0.0.1:3306/java1_bookstore" />
        <property name="username" value="root" />
        <property name="password" value="1234" />
    </bean>

    <!-- Mybatis 설정 -->
    <bean id="sqlSessionFactoryBean" class="org.mybatis.spring.SqlSessionFactoryBean">
        <property name="dataSource" ref="dataSource"/>
        <property name="configLocation" value="classpath:mybatis-context.xml"/>
    </bean>

    <!-- Mybatis-Spring 설정 -->
    <bean id="sqlSessionTemplate" class="org.mybatis.spring.SqlSessionTemplate">
        <constructor-arg ref="sqlSessionFactoryBean"/>
    </bean>

</beans>
```



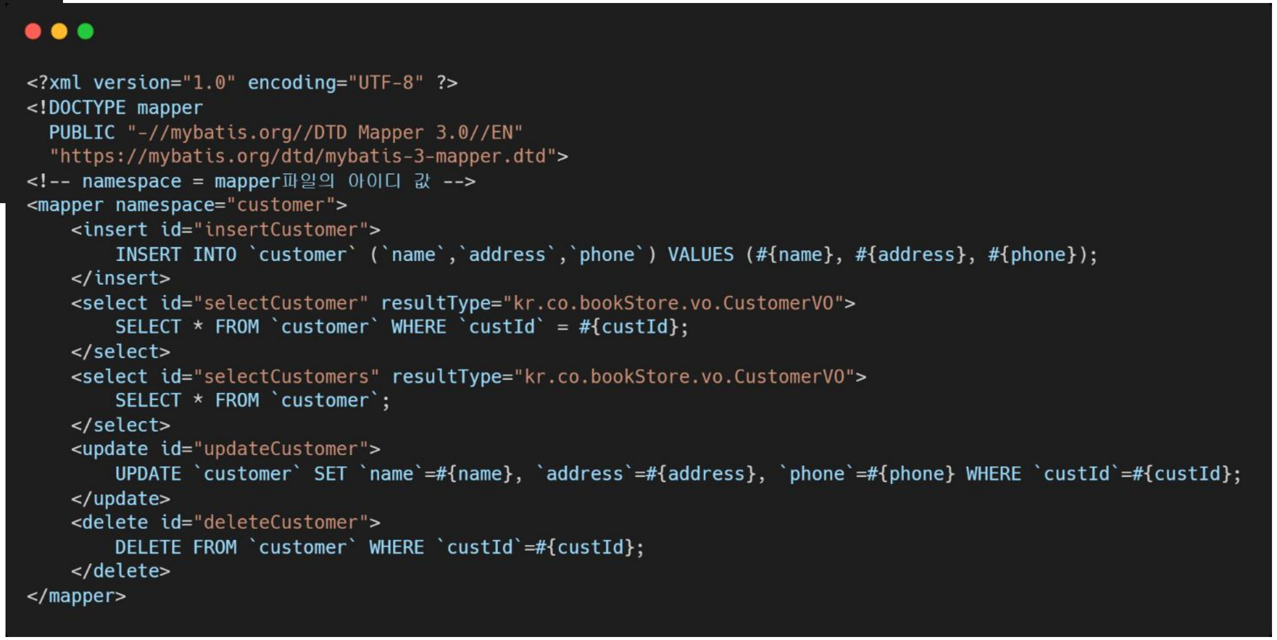
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE configuration
  PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
  "https://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
  <mappers>
    <mapper resource="./mappers/book.xml"/>
    <mapper resource="./mappers/customer.xml"/>
  </mappers>
</configuration>
```

스프링과 Mybatis 연결, Mybatis 설정

Part 3 기능구현



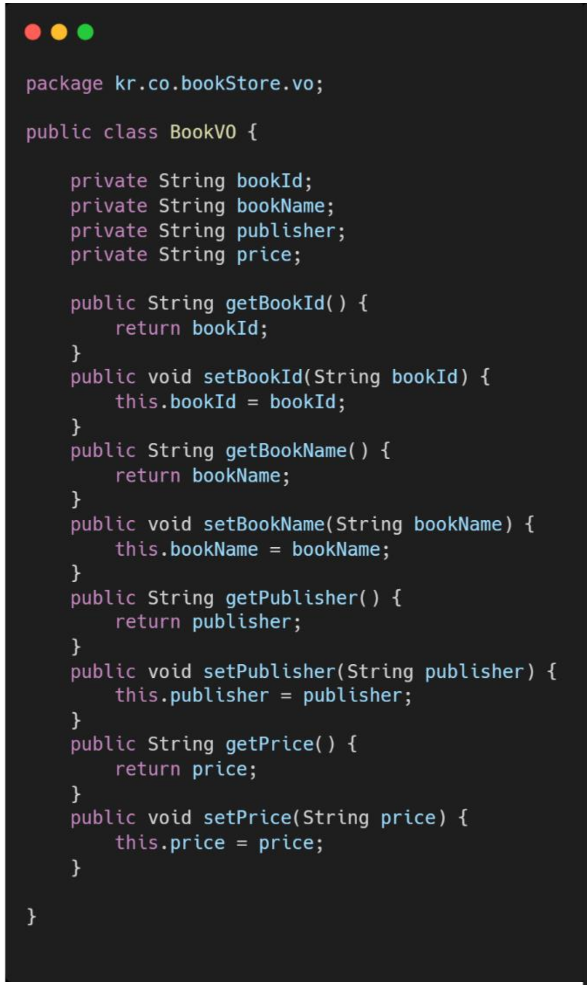
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
  "https://mybatis.org/dtd/mybatis-3-mapper.dtd">
<!-- namespace = mapper파일의 아이디 값 -->
<mapper namespace="book">
    <insert id="insertBook">
        INSERT INTO `book` (`bookName`,`publisher`,`price`) VALUES (#{bookName}, #{publisher}, #{price});
    </insert>
    <select id="selectBook" resultType="kr.co.bookStore.vo.BookVO">
        SELECT * FROM `book` WHERE `bookId` = #{bookId};
    </select>
    <select id="selectBooks" resultType="kr.co.bookStore.vo.BookVO">
        SELECT * FROM `book`;
    </select>
    <update id="updateBook">
        UPDATE `book` SET `bookName`=#{bookName}, `publisher`=#{publisher}, `price`=#{price} WHERE `bookId`=#
{bookId};
    </update>
    <delete id="deleteBook">
        DELETE FROM `book` WHERE `bookId`=#{bookId};
    </delete>
</mapper>
```



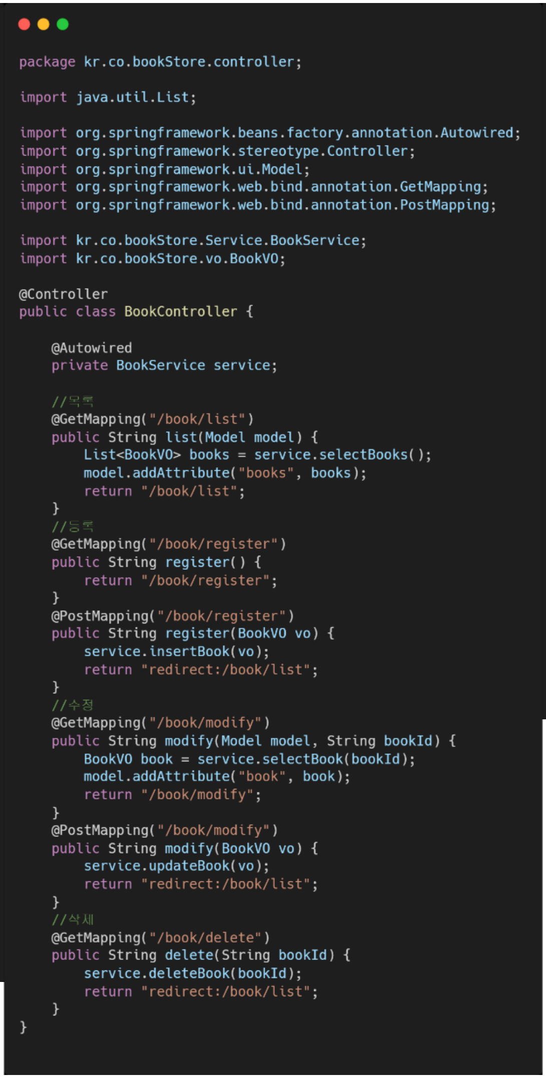
```xml
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
  "https://mybatis.org/dtd/mybatis-3-mapper.dtd">
<!-- namespace = mapper파일의 아이디 값 -->
<mapper namespace="customer">
    <insert id="insertCustomer">
        INSERT INTO `customer` (`name`,`address`,`phone`) VALUES (#{name}, #{address}, #{phone});
    </insert>
    <select id="selectCustomer" resultType="kr.co.bookStore.vo.CustomerVO">
        SELECT * FROM `customer` WHERE `custId` = #{custId};
    </select>
    <select id="selectCustomers" resultType="kr.co.bookStore.vo.CustomerVO">
        SELECT * FROM `customer`;
    </select>
    <update id="updateCustomer">
        UPDATE `customer` SET `name`=#{name}, `address`=#{address}, `phone`=#{phone} WHERE `custId`=#{custId};
    </update>
    <delete id="deleteCustomer">
        DELETE FROM `customer` WHERE `custId`=#{custId};
    </delete>
</mapper>
```

Mybatis SQL 설정

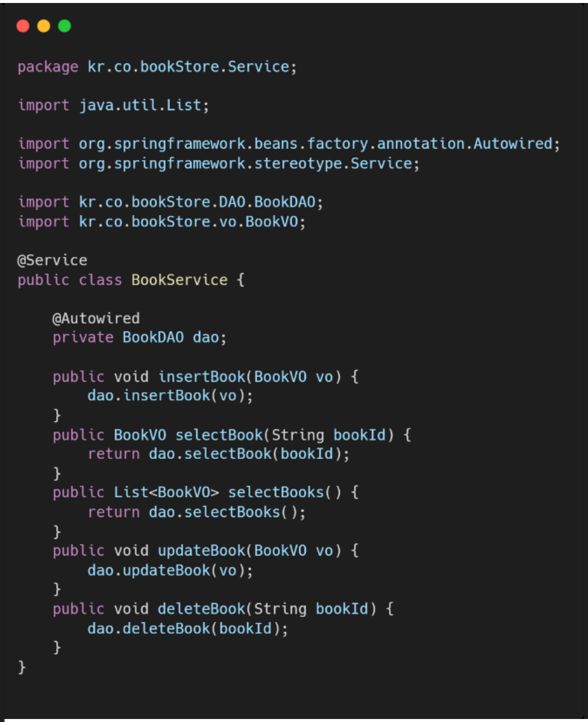
Part 3 기능구현



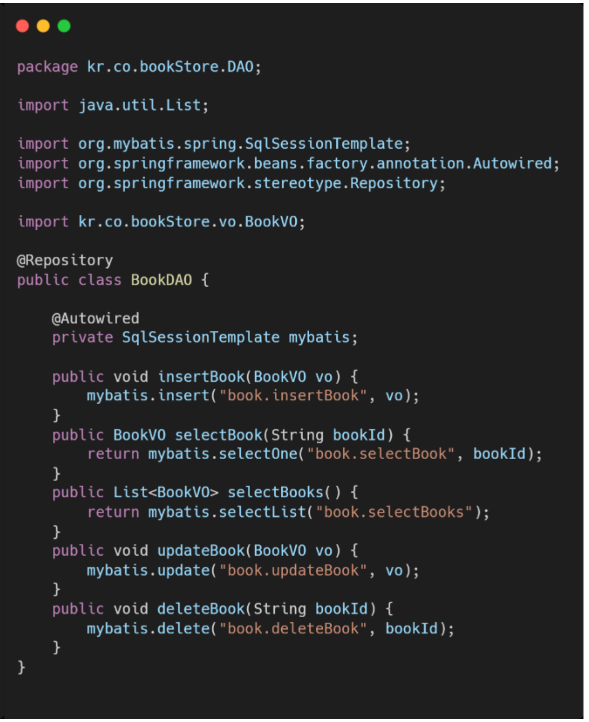
```
package kr.co.bookStore.vo;

public class BookVO {

    private String bookId;
    private String bookName;
    private String publisher;
    private String price;

    public String getBookId() {
        return bookId;
    }
    public void setBookId(String bookId) {
        this.bookId = 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 String getPrice() {
        return price;
    }
    public void setPrice(String price) {
        this.price = price;
    }

}
```



```
package kr.co.bookStore.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;

import kr.co.bookStore.Service.BookService;
import kr.co.bookStore.vo.BookVO;

@Controller
public class BookController {

    @Autowired
    private BookService service;

    //목록
    @GetMapping("/book/list")
    public String list(Model model) {
        List<BookVO> books = service.selectBooks();
        model.addAttribute("books", books);
        return "/book/list";
    }
    //등록
    @GetMapping("/book/register")
    public String register() {
        return "/book/register";
    }
    @PostMapping("/book/register")
    public String register(BookVO vo) {
        service.insertBook(vo);
        return "redirect:/book/list";
    }
    //수정
    @GetMapping("/book/modify")
    public String modify(Model model, String bookId) {
        BookVO book = service.selectBook(bookId);
        model.addAttribute("book", book);
        return "/book/modify";
    }
    @PostMapping("/book/modify")
    public String modify(BookVO vo) {
        service.updateBook(vo);
        return "redirect:/book/list";
    }
    //삭제
    @GetMapping("/book/delete")
    public String delete(String bookId) {
        service.deleteBook(bookId);
        return "redirect:/book/list";
    }
}
```



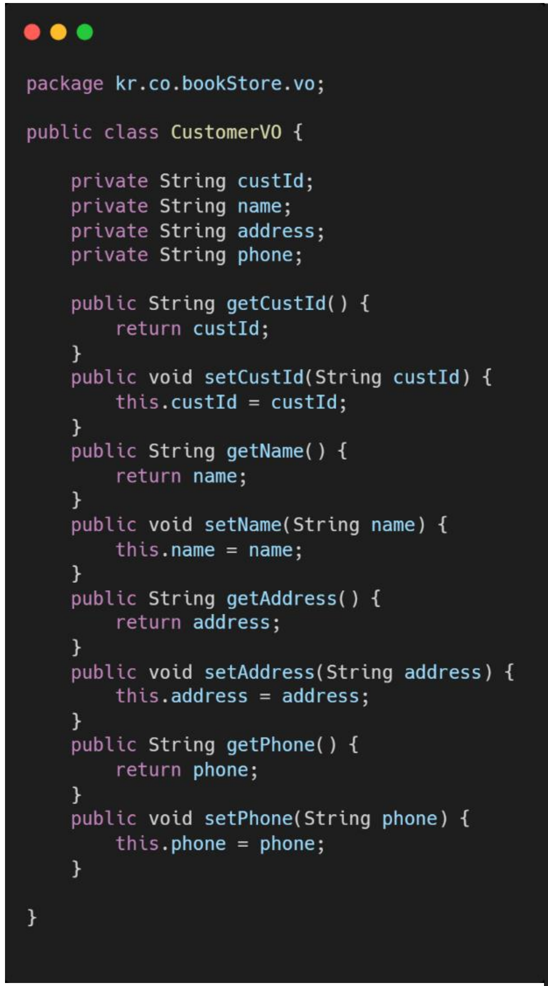
```
package kr.co.bookStore.Service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import kr.co.bookStore.DAO.BookDAO;
import kr.co.bookStore.vo.BookVO;

@Service
public class BookService {

    @Autowired
    private BookDAO dao;

    public void insertBook(BookVO vo) {
        dao.insertBook(vo);
    }
    public BookVO selectBook(String bookId) {
        return dao.selectBook(bookId);
    }
    public List<BookVO> selectBooks() {
        return dao.selectBooks();
    }
    public void updateBook(BookVO vo) {
        dao.updateBook(vo);
    }
    public void deleteBook(String bookId) {
        dao.deleteBook(bookId);
    }
}
```



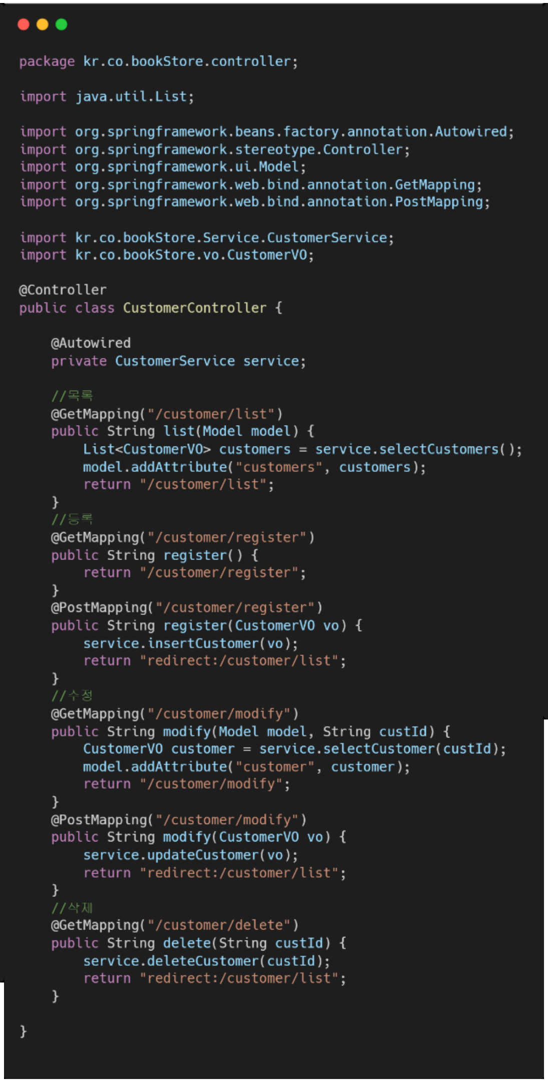
```
package kr.co.bookStore.DAO;

import java.util.List;

import org.mybatis.spring.SqlSessionTemplate;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;

import kr.co.bookStore.vo.BookVO;

@Repository
public class BookDAO {

    @Autowired
    private SqlSessionTemplate mybatis;

    public void insertBook(BookVO vo) {
        mybatis.insert("book.insertBook", vo);
    }
    public BookVO selectBook(String bookId) {
        return mybatis.selectOne("book.selectBook", bookId);
    }
    public List<BookVO> selectBooks() {
        return mybatis.selectList("book.selectBooks");
    }
    public void updateBook(BookVO vo) {
        mybatis.update("book.updateBook", vo);
    }
    public void deleteBook(String bookId) {
        mybatis.delete("book.deleteBook", bookId);
    }
}
```

Book VO, Controller, Service, DAO

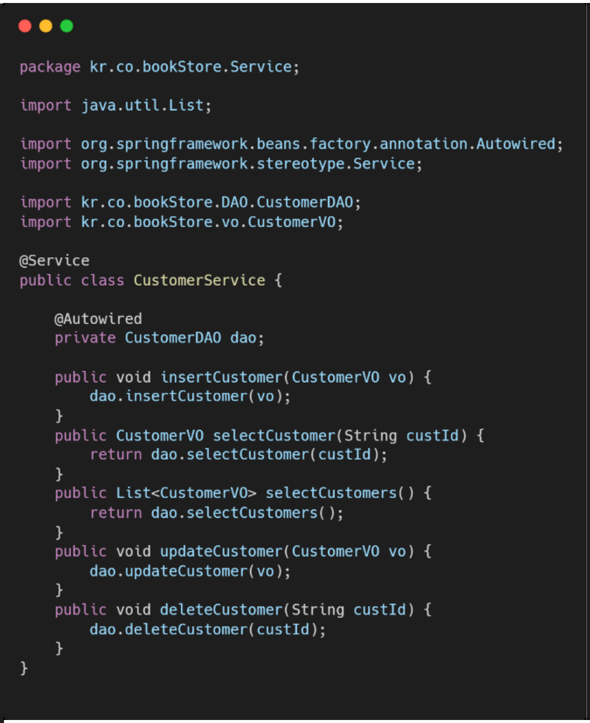
Part 3 기능구현



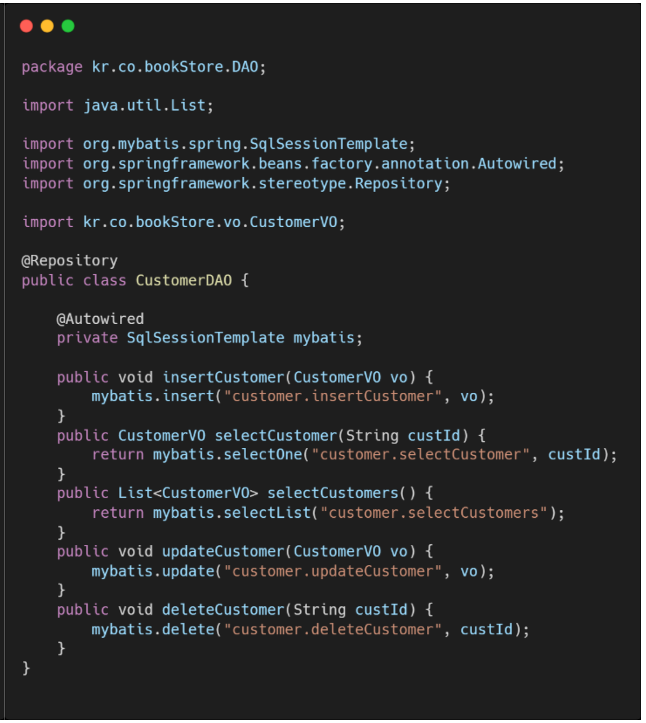
```java
package kr.co.bookStore.vo;

public class CustomerVO {

    private String custId;
    private String name;
    private String address;
    private String phone;

    public String getCustId() {
        return custId;
    }
    public void setCustId(String custId) {
        this.custId = 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;
    }

}
```



```java
package kr.co.bookStore.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;

import kr.co.bookStore.Service.CustomerService;
import kr.co.bookStore.vo.CustomerVO;

@Controller
public class CustomerController {

    @Autowired
    private CustomerService service;

    //목록
    @GetMapping("/customer/list")
    public String list(Model model) {
        List<CustomerVO> customers = service.selectCustomers();
        model.addAttribute("customers", customers);
        return "/customer/list";
    }
    //등록
    @GetMapping("/customer/register")
    public String register() {
        return "/customer/register";
    }
    @PostMapping("/customer/register")
    public String register(CustomerVO vo) {
        service.insertCustomer(vo);
        return "redirect:/customer/list";
    }
    //수정
    @GetMapping("/customer/modify")
    public String modify(Model model, String custId) {
        CustomerVO customer = service.selectCustomer(custId);
        model.addAttribute("customer", customer);
        return "/customer/modify";
    }
    @PostMapping("/customer/modify")
    public String modify(CustomerVO vo) {
        service.updateCustomer(vo);
        return "redirect:/customer/list";
    }
    //삭제
    @GetMapping("/customer/delete")
    public String delete(String custId) {
        service.deleteCustomer(custId);
        return "redirect:/customer/list";
    }

}
```



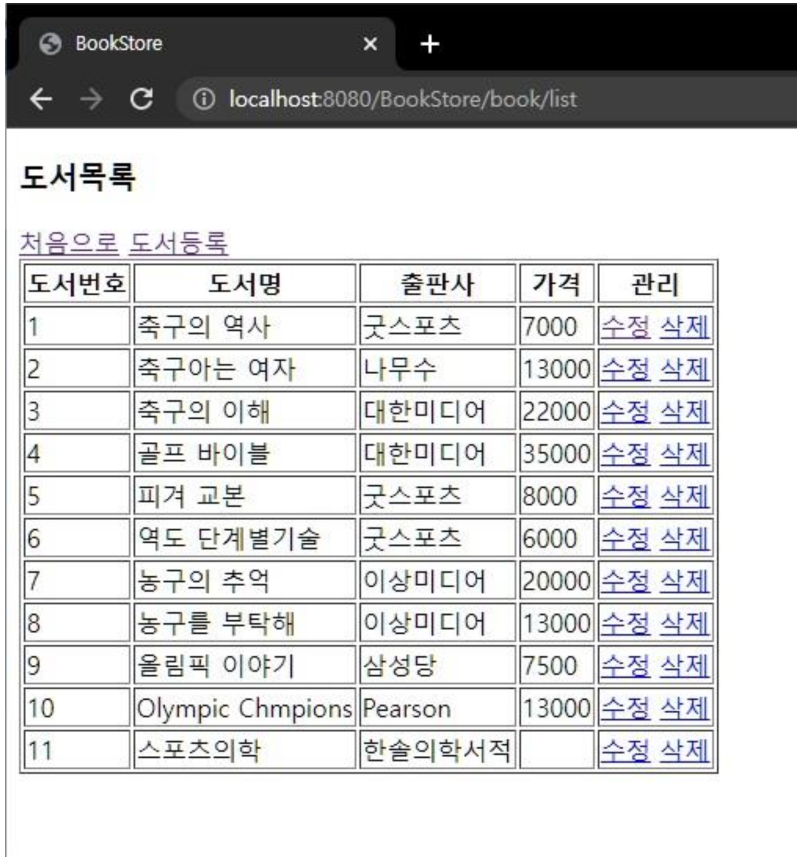
```java
package kr.co.bookStore.Service;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;

import kr.co.bookStore.DAO.CustomerDAO;
import kr.co.bookStore.vo.CustomerVO;

@Service
public class CustomerService {

    @Autowired
    private CustomerDAO dao;

    public void insertCustomer(CustomerVO vo) {
        dao.insertCustomer(vo);
    }
    public CustomerVO selectCustomer(String custId) {
        return dao.selectCustomer(custId);
    }
    public List<CustomerVO> selectCustomers() {
        return dao.selectCustomers();
    }
    public void updateCustomer(CustomerVO vo) {
        dao.updateCustomer(vo);
    }
    public void deleteCustomer(String custId) {
        dao.deleteCustomer(custId);
    }
}
```



```java
package kr.co.bookStore.DAO;

import java.util.List;

import org.mybatis.spring.SqlSessionTemplate;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;

import kr.co.bookStore.vo.CustomerVO;

@Repository
public class CustomerDAO {

    @Autowired
    private SqlSessionTemplate mybatis;

    public void insertCustomer(CustomerVO vo) {
        mybatis.insert("customer.insertCustomer", vo);
    }
    public CustomerVO selectCustomer(String custId) {
        return mybatis.selectOne("customer.selectCustomer", custId);
    }
    public List<CustomerVO> selectCustomers() {
        return mybatis.selectList("customer.selectCustomers");
    }
    public void updateCustomer(CustomerVO vo) {
        mybatis.update("customer.updateCustomer", vo);
    }
    public void deleteCustomer(String custId) {
        mybatis.delete("customer.deleteCustomer", custId);
    }
}
```

Customer VO, Controller, Service, DAO

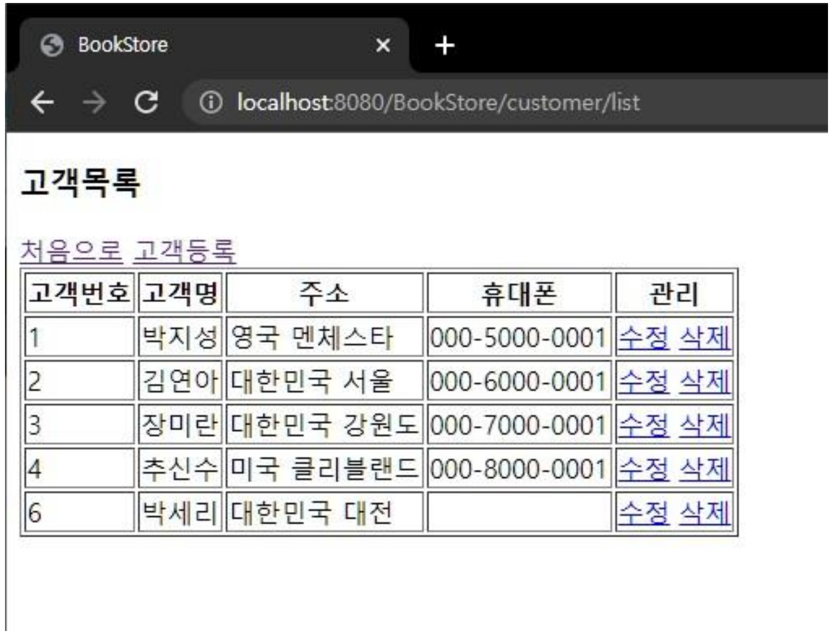
Part 4 실행



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



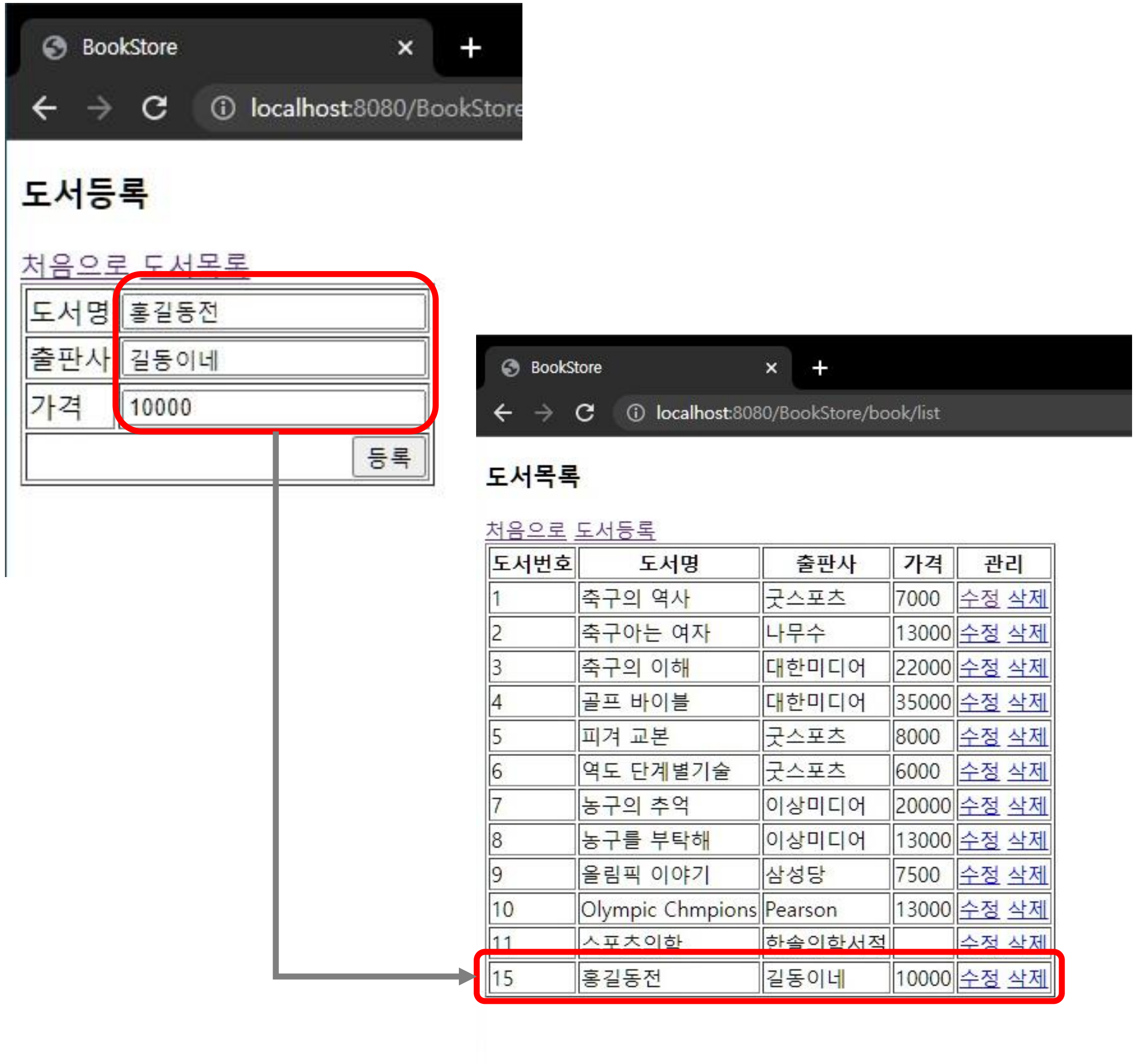
고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 멘체스타	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

도서 목록, 고객 목록

Part 4 실행



도서등록

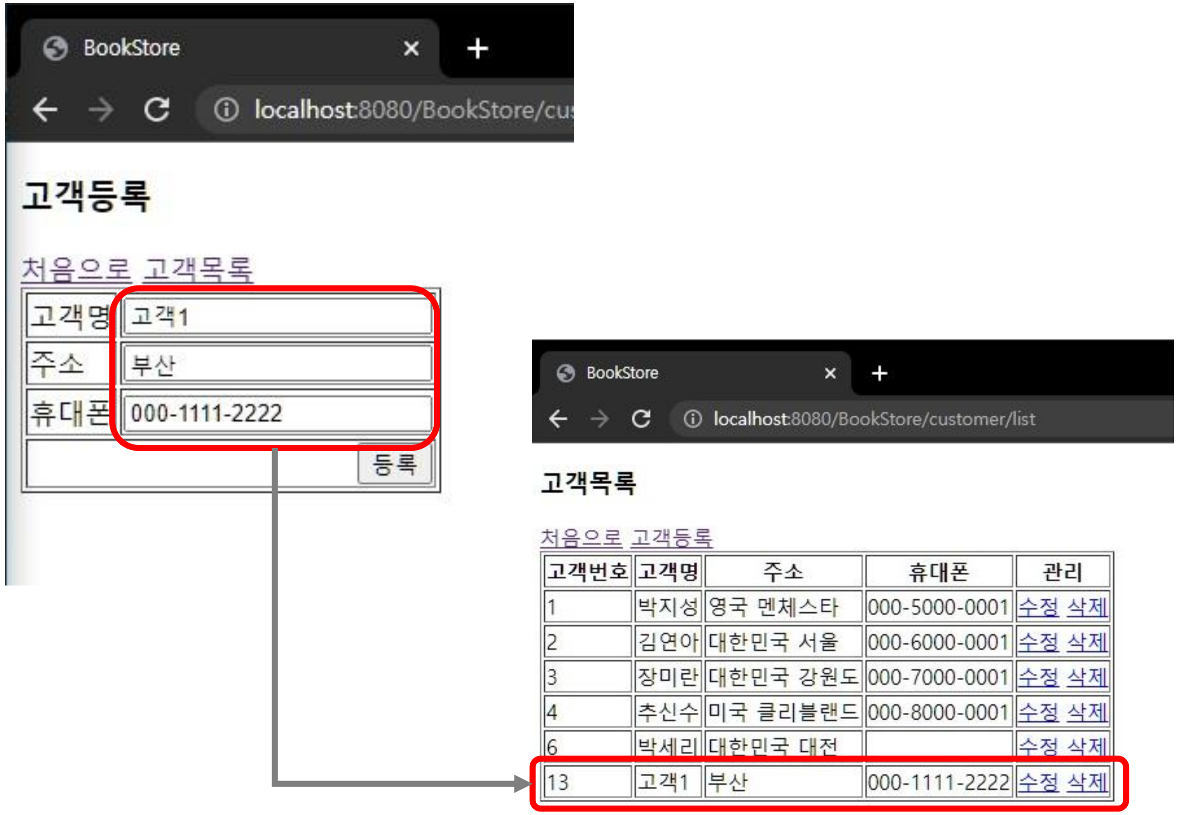
처음으로 도서목록

도서명	홍길동전
출판사	길동이네
가격	10000
	등록

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
15	홍길동전	길동이네	10000	수정 삭제



BookStore
localhost:8080/BookStore/cu

고객등록

처음으로 고객목록

고객명	고객1
주소	부산
휴대폰	000-1111-2222
	등록

BookStore
localhost:8080/BookStore/customer/list

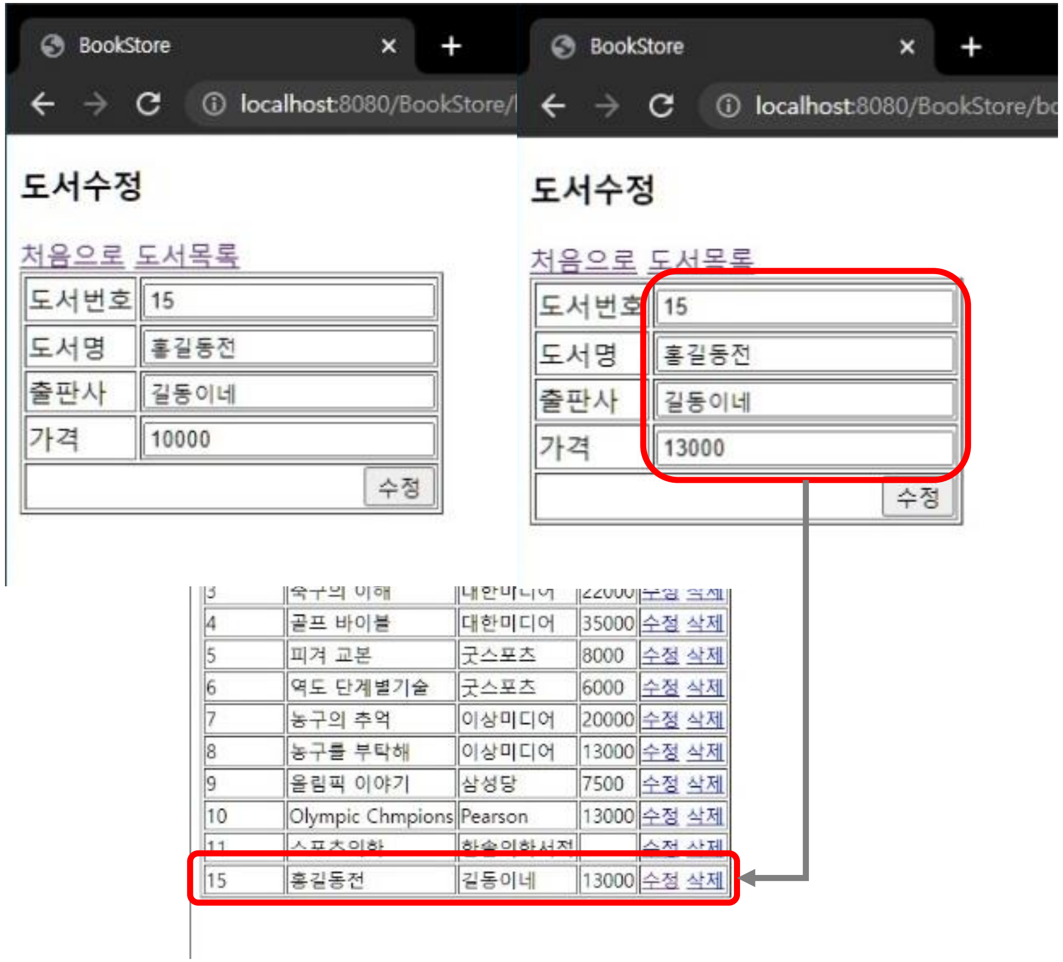
고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 멘체스타	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
13	고객1	부산	000-1111-2222	수정 삭제

도서 등록, 고객 등록

Part 4 실행



도서수정

처음으로 도서목록

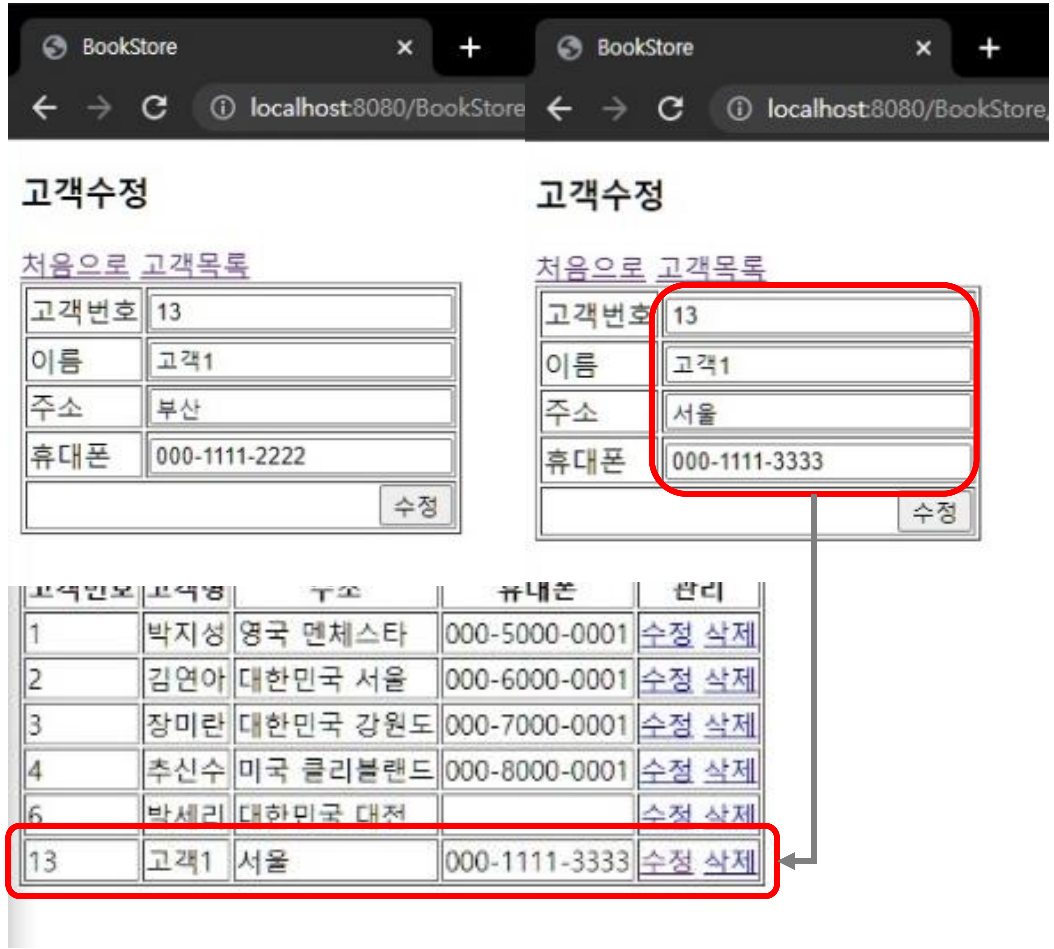
도서번호	15
도서명	홍길동전
출판사	길동이네
가격	10000
	수정

도서수정

처음으로 도서목록

도서번호	15
도서명	홍길동전
출판사	길동이네
가격	13000
	수정

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
15	홍길동전	길동이네	13000	수정 삭제



고객수정

처음으로 고객목록

고객번호	13
이름	고객1
주소	부산
휴대폰	000-1111-2222
	수정

고객수정

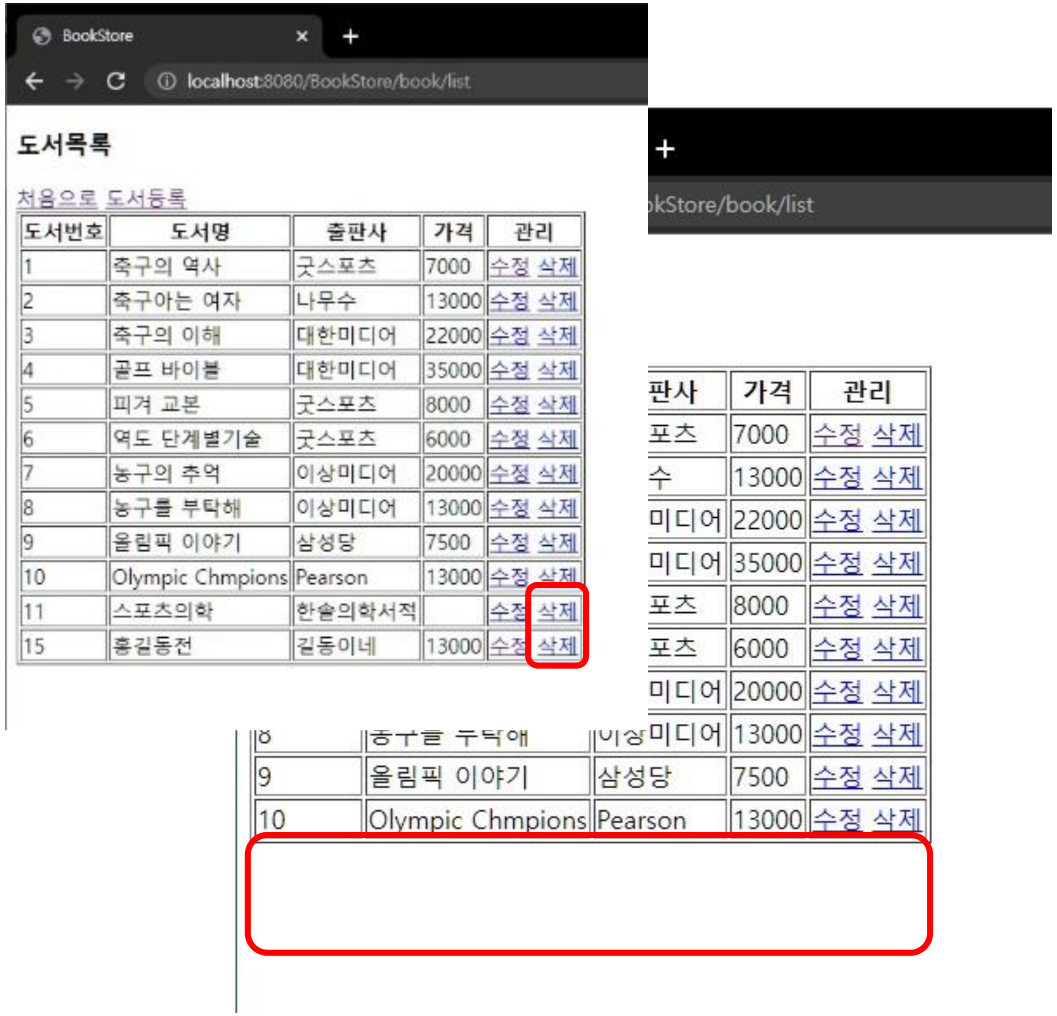
처음으로 고객목록

고객번호	13
이름	고객1
주소	서울
휴대폰	000-1111-3333
	수정

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 멘체스타	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
13	고객1	서울	000-1111-3333	수정 삭제

도서 수정, 고객 수정

Part 4 실행



BookStore

localhost:8080/BookStore/book/list

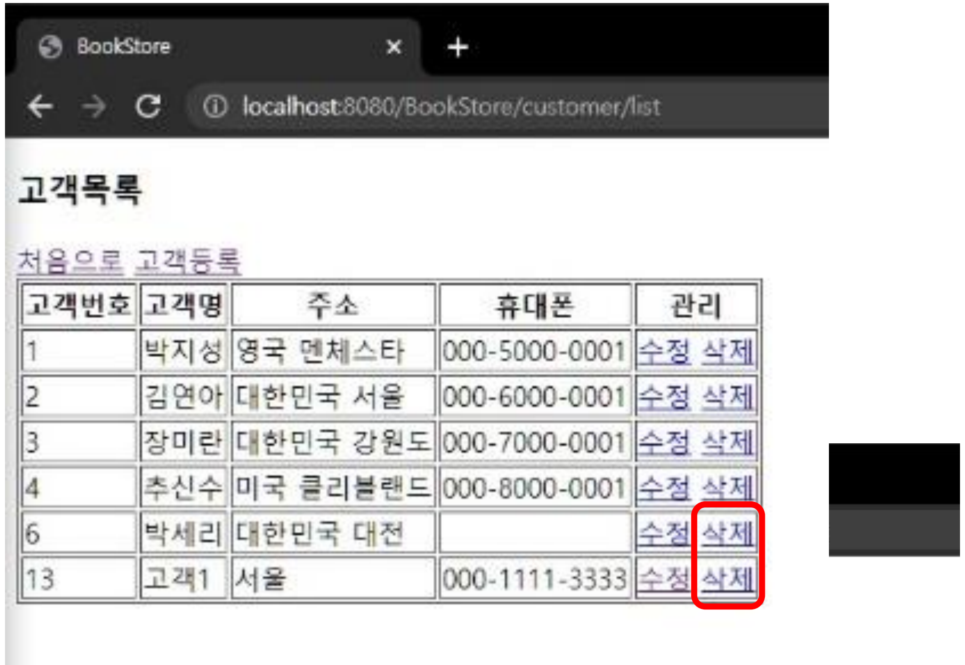
도서목록

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
15	홍길동전	길동이네	13000	수정 삭제

kStore/book/list

		판사	가격	관리
		포츠	7000	수정 삭제
		수	13000	수정 삭제
		미디어	22000	수정 삭제
		미디어	35000	수정 삭제
		포츠	8000	수정 삭제
		포츠	6000	수정 삭제
		미디어	20000	수정 삭제
8	농구를 부탁해	이상미디어	13000	수정 삭제
9	올림픽 이야기	삼성당	7500	수정 삭제
10	Olympic Chmpions	Pearson	13000	수정 삭제



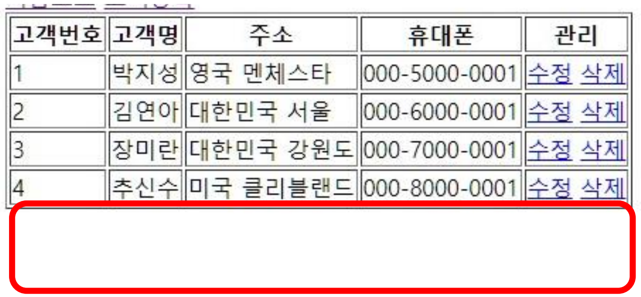
BookStore

localhost:8080/BookStore/customer/list

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스타	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
13	고객1	서울	000-1111-3333	수정 삭제



고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 맨체스타	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

도서 삭제, 고객 삭제