

모던 웹 애플리케이션 개발2

이름 : 공민혁
반 : JAVA1 오전반

1. 프로젝트 생성 및 구성

2. 화면구현

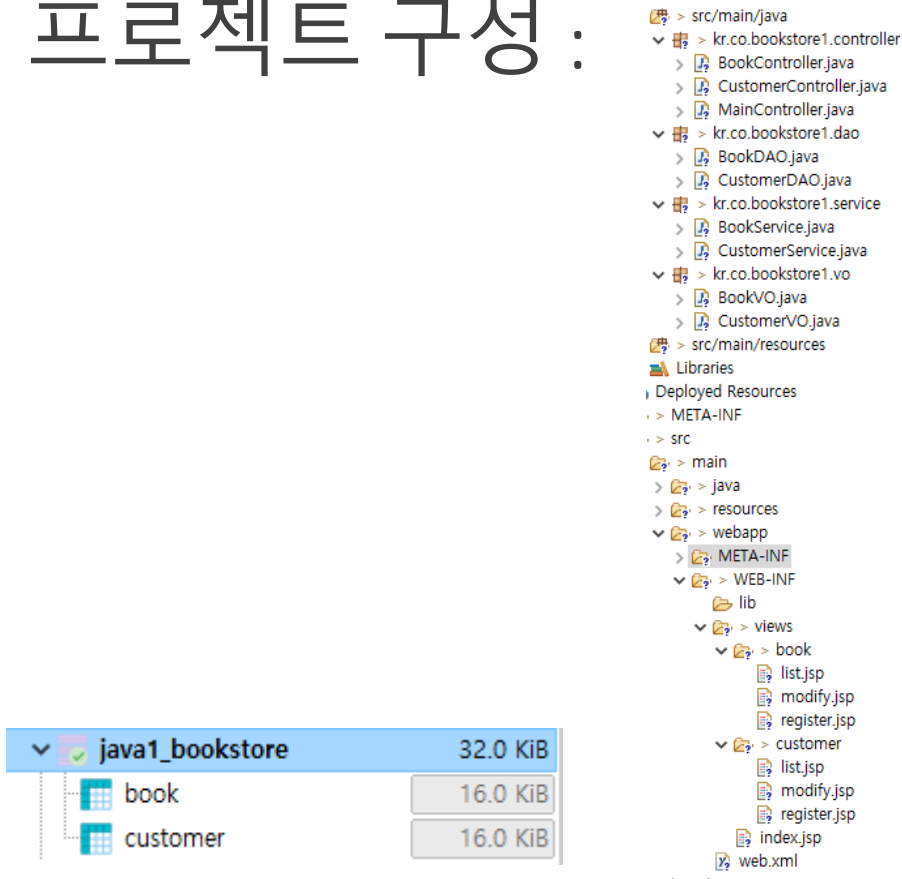
3. 기능구현

4. 실행

1. 프로젝트 생성 및 구성

1. 프로젝트 생성 및 구성

프로젝트 구성 :



프로젝트 이름 : BookStore

WAS : Tomcat 9

DB : java1_bookstore

Framework : Spring, Mybatis

Tools : Eclipse, WrokBench

2. 화면구현

2. 화면구현 (main, index)

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>index</title>
</head>
<body>
  <h3>BookStore</h3>

  <a href="/BookStore1/book/list">도서 목록</a>
  <a href="/BookStore1/customer/list">고객 목록</a>
</body>
</html>
```

2. 화면구현(book)

```
<%@ 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>book::list</title>
  </head>
  <body>
    <h3>도서목록</h3>

    <a href="/BookStore1/">처음으로</a>
    <a href="/BookStore1/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 var="book" items="${books}">
        <tr>
          <td>${book.bookId}</td>
          <td>${book.bookName}</td>
          <td>${book.pubName}</td>
          <td>${book.price}</td>
          <td>
            <a href="/BookStore1/book/modify?bookName=${book.bookName}">수정</a>
            <a href="/BookStore1/book/delete?bookName=${book.bookName}">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>book::register</title>
  </head>
  <body>
    <h3>도서등록</h3>

    <a href="/BookStore1/">처음으로</a>
    <a href="/BookStore1/book/list">도서목록</a>

    <form action="/BookStore1/book/register" 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="pubName"/></td>
        </tr>
        <tr>
          <td>가격</td>
          <td><input type="number" name="price"/></td>
        </tr>
        <tr>
          <td colspan="2" align="right">
            <input type="submit" value="등록하기"/>
          </td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

```
<%@ 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>book::modify</title>
  </head>
  <body>
    <h3>도서수정</h3>

    <a href="/BookStore1/">처음으로</a>
    <a href="/BookStore1/book/list">도서목록</a>

    <form action="/BookStore1/book/modify" method="post">
      <table border="1">
        <tr>
          <td>도서번호</td>
          <td><input type="text" name="bookId" readonly value="${book.bookId}"/></td>
        </tr>
        <tr>
          <td>도서명</td>
          <td><input type="text" name="bookName" value="${book.bookName}"/></td>
        </tr>
        <tr>
          <td>출판사</td>
          <td><input type="text" name="pubName" value="${book.pubName}"/></td>
        </tr>
        <tr>
          <td>가격</td>
          <td><input type="number" name="price" value="${book.price}"/></td>
        </tr>
        <tr>
          <td colspan="2" align="right">
            <input type="submit" value="수정"/>
          </td>
        </tr>
      </table>
    </form>
  </body>
</html>
```

2. 화면구현(customer)

```
<%@ 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>customer::list</title>
</head>
<body>
<h3>고객목록</h3>

<a href="/BookStore1/">처음으로</a>
<a href="/BookStore1/customer/register">고객등록</a>

<table border="1">
<tr>
<th>고객번호</th>
<th>고객명</th>
<th>주소</th>
<th>휴대폰</th>
<th>관리</th>
</tr>
<tr>
<td>
<c:forEach var="customer" items="${customers}">
<tr>
<td>${customer.cusId}</td>
<td>${customer.cusName}</td>
<td>${customer.cusAddr}</td>
<td>${customer.cusHp}</td>
<td>
<a href="/BookStore1/customer/modify?cusName=${customer.cusName}">수정</a>
<a href="/BookStore1/customer/delete?cusName=${customer.cusName}">삭제</a>
</td>
</tr>
</c:forEach>
</td>
</tr>
</table>
</body>
</html>
```

```
<%@ 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>customer::modify</title>
</head>
<body>
<h3>고객수정</h3>

<a href="/BookStore1/">처음으로</a>
<a href="/BookStore1/customer/list">고객목록</a>
<form action="/BookStore1/customer/modify" method="post">
<table border="1">
<tr>
<td>고객번호</td>
<td><input type="text" name="cusId" readonly value="${customer.cusId}"/></td>
</tr>
<tr>
<td>고객명</td>
<td><input type="text" name="cusName" value="${customer.cusName}"/></td>
</tr>
<tr>
<td>주소</td>
<td><input type="text" name="cusAddr" value="${customer.cusAddr}"/></td>
</tr>
<tr>
<td>휴대폰</td>
<td><input type="text" name="cusHp" value="${customer.cusHp}"/></td>
</tr>
<tr>
<td colspan="2" align="right">
<input type="submit" value="수정"/>
</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>customer::register</title>
</head>
<body>
<h3>고객등록</h3>

<a href="/BookStore1/">처음으로</a>
<a href="/BookStore1/customer/list">고객목록</a>

<form action="/BookStore1/customer/register" method="post">
<table border="1">
<tr>
<td>고객명</td>
<td><input type="text" name="cusName"/></td>
</tr>
<tr>
<td>주소</td>
<td><input type="text" name="cusAddr"/></td>
</tr>
<tr>
<td>휴대폰</td>
<td><input type="text" name="cusHp"/></td>
</tr>
<tr>
<td colspan="2" align="right">
<input type="submit" value="등록하기"/>
</td>
</tr>
</table>
</form>
</body>
</html>
```


3. 기능구현

3. 기능구현 (maven 설정)

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xs
<modelVersion>4.0.0</modelVersion>
<groupId>kr.co.bookstore1</groupId>
<artifactId>BookStore1</artifactId>
<packaging>war</packaging>
<version>0.0.1-SNAPSHOT</version>
<name>BookStore1 Maven Webapp</name>
<url>http://maven.apache.org</url>
<dependencies>
<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.aspectj</groupId>
<artifactId>aspectjweaver</artifactId>
<version>1.9.19</version>
</dependency>
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-webmvc</artifactId>
<version>5.3.24</version>
</dependency>
<dependency>
<groupId>org.mybatis</groupId>
<artifactId>mybatis</artifactId>
<version>3.5.11</version>
</dependency>
<dependency>
<groupId>org.mybatis</groupId>
<artifactId>mybatis-spring</artifactId>
<version>2.1.0</version>
</dependency>
```

```
<dependency>
<groupId>org.springframework</groupId>
<artifactId>spring-jdbc</artifactId>
<version>5.3.24</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>javax.servlet</groupId>
<artifactId>jstl</artifactId>
<version>1.2</version>
</dependency>
</dependencies>
<build>
<finalName>BookStore1</finalName>
</build>
</project>
```

```
<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>
<servlet-mapping>
<servlet-name>app</servlet-name>
<url-pattern>/</url-pattern>
</servlet-mapping>
<!-- 인코딩 필터 -->
<filter>
<filter-name>characterEncodingFilter</filter-name>
<filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>
<init-param>
<param-name>encoding</param-name>
<param-value>UTF-8</param-value>
</init-param>
</filter>
<filter-mapping>
<filter-name>characterEncodingFilter</filter-name>
<url-pattern>*</url-pattern>
</filter-mapping>
```

Pom.xml , web.xml

3. 기능구현 (mybatis 설정)

```
<?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
    http://www.springframework.org/schema/beans/spring-beans.xsd
    http://www.springframework.org/schema/context
    http://www.springframework.org/schema/context/spring-context.xsd
    http://www.springframework.org/schema/mvc
    http://www.springframework.org/schema/mvc/spring-mvc.xsd">

  <!-- 컴포넌트 등록을 위한 태그선언 -->
  <context:component-scan base-package="kr.co.bookstore1"/>

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

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

  <!-- 데이터베이스(커넥션풀) 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>
    <property name="configLocation" value="classpath:mybatis-context.xml"></property>
  </bean>

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

</beans>
```

```
<?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>
```

Spring , mybatis 설정 xml

3. 기능구현 (mapper)

```
<?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">
<mapper namespace="book">
  <insert id="insertBook">
    INSERT INTO `book` VALUES (#{bookId}, #{bookName}, #{pubName}, #{price});
  </insert>
  <select id="selectBook" resultType="kr.co.bookstore1.vo.BookVO">
    SELECT * FROM `book` WHERE `bookName`=#{bookName};
  </select>
  <select id="selectBooks" resultType="kr.co.bookstore1.vo.BookVO">
    SELECT * FROM `book`;
  </select>
  <update id="updateBook">
    UPDATE `book` SET
      `bookName`=#{bookName},
      `pubName`=#{pubName},
      `price`=#{price}
    WHERE
      `bookId`=#{bookId};
  </update>
  <delete id="deleteBook">
    DELETE FROM `book` WHERE `bookName`=#{bookName};
  </delete>
```

book

```
<?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">
<mapper namespace="customer">
  <insert id="insertCustomer">
    INSERT INTO `customer` VALUES (#{cusId}, #{cusName}, #{cusAddr}, #{cusHp});
  </insert>
  <select id="selectCustomer" resultType="kr.co.bookstore1.vo.CustomerVO">
    SELECT * FROM `customer` WHERE `cusName`=#{cusName};
  </select>
  <select id="selectCustomers" resultType="kr.co.bookstore1.vo.CustomerVO">
    SELECT * FROM `customer`;
  </select>
  <update id="updateCustomer">
    UPDATE `customer` SET
      `cusName`=#{cusName},
      `cusAddr`=#{cusAddr},
      `cusHp`=#{cusHp}
    WHERE
      `cusId`=#{cusId};
  </update>
  <delete id="deleteCustomer">
    DELETE FROM `customer` WHERE `cusName`=#{cusName};
  </delete>
</mapper>
```

customer

3. 기능구현(controller)

```
package kr.co.bookstore1.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";
    }

}
```

index

```
@Autowired
private BookService service;

@GetMapping("/book/list")
public String list(Model model) {

    List<BookVO> books = service.selectBooks();

    model.addAttribute("books", books);

    return "/book/list";
}

@GetMapping("/book/register")
public String register() {
    return "/book/register";
}

@PostMapping("/book/register")
public String register(BookVO vo) {
    service.insertBook(vo);
    return "redirect:/book/list";
}

@GetMapping("/book/modify")
public String modify(Model model, String bookName)

    BookVO book = service.selectBook(bookName);
    model.addAttribute("book", book);

    return "/book/modify";
}

@PostMapping("/book/modify")
public String modify(BookVO vo) {

    service.updateBook(vo);

    return "redirect:/book/list";
}

@GetMapping("/book/delete")
public String delete(String bookName) {

    service.deleteBook(bookName);

    return "redirect:/book/list";
}
```

book

```
@Autowired
private CustomerService service;

@GetMapping("/customer/list")
public String list(Model model) {

    List<CustomerVO> customers = service.selectCustomers();

    model.addAttribute("customers", customers);

    return "/customer/list";
}

@GetMapping("/customer/register")
public String register() {
    return "/customer/register";
}

@PostMapping("/customer/register")
public String register(CustomerVO vo) {
    service.insertCustomer(vo);
    return "redirect:/customer/list";
}

@GetMapping("/customer/modify")
public String modify(Model model, String cusName) {

    CustomerVO customer = service.selectCustomer(cusName);
    model.addAttribute("customer", customer);

    return "/customer/modify";
}

@PostMapping("/customer/modify")
public String modify(CustomerVO vo) {

    service.updateCustomer(vo);

    return "redirect:/customer/list";
}

@GetMapping("/customer/delete")
public String delete(String cusName) {

    service.deleteCustomer(cusName);

    return "redirect:/customer/list";
}
```

customer

3. 기능구현(DAO)

```
package kr.co.bookstore1.dao;

import java.util.List;

@Repository
public class BookDAO {

    @Autowired
    private SqlSessionTemplate mybatis;

    public void insertBook(BookVO vo) {
        mybatis.insert("book.insertBook", vo);
    }

    public BookVO selectBook(String bookName) {
        return mybatis.selectOne("book.selectBook", bookName);
    }

    public List<BookVO> selectBooks() {
        return mybatis.selectList("book.selectBooks");
    }

    public void updateBook(BookVO vo) {
        mybatis.update("book.updateBook", vo);
    }

    public void deleteBook(String bookName) {
        mybatis.delete("book.deleteBook", bookName);
    }
}
```

book

```
package kr.co.bookstore1.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.bookstore1.vo.CustomerVO;

@Repository
public class CustomerDAO {

    @Autowired
    private SqlSessionTemplate mybatis;

    public void insertCustomer(CustomerVO vo) {
        mybatis.insert("customer.insertCustomer", vo);
    }

    public CustomerVO selectCustomer(String cusName) {
        return mybatis.selectOne("customer.selectCustomer", cusName);
    }

    public List<CustomerVO> selectCustomers() {
        return mybatis.selectList("customer.selectCustomers");
    }

    public void updateCustomer(CustomerVO vo) {
        mybatis.update("customer.updateCustomer", vo);
    }

    public void deleteCustomer(String cusName) {
        mybatis.delete("customer.deleteCustomer", cusName);
    }
}
```

customer

3. 기능구현(service)

```
package kr.co.bookstore1.service;

import java.util.List;

@Service
public class BookService {

    @Autowired
    private BookDAO dao;

    public void insertBook(BookVO vo) {
        dao.insertBook(vo);
    }

    public BookVO selectBook(String bookName) {
        return dao.selectBook(bookName);
    }

    public List<BookVO> selectBooks() {
        return dao.selectBooks();
    }

    public void updateBook(BookVO vo) {
        dao.updateBook(vo);
    }

    public void deleteBook(String bookName) {
        dao.deleteBook(bookName);
    }
}
```

book

```
package kr.co.bookstore1.service;

import java.util.List;

@Service
public class CustomerService {

    @Autowired
    private CustomerDAO dao;

    public void insertCustomer(CustomerVO vo) {
        dao.insertCustomer(vo);
    }

    public CustomerVO selectCustomer(String cusName) {
        return dao.selectCustomer(cusName);
    }

    public List<CustomerVO> selectCustomers() {
        return dao.selectCustomers();
    }

    public void updateCustomer(CustomerVO vo) {
        dao.updateCustomer(vo);
    }

    public void deleteCustomer(String cusName) {
        dao.deleteCustomer(cusName);
    }
}
```

customer

3. 기능구현(vo)

```
package kr.co.bookstore1.vo;

public class BookVO {

    private int bookId;
    private String bookName;
    private String pubName;
    private int price;

    public int getBookId() {
        return bookId;
    }
    public void setBookId(int bookId) {
        this.bookId = bookId;
    }
    public String getBookName() {
        return bookName;
    }
    public void setBookName(String bookName) {
        this.bookName = bookName;
    }
    public String getPubName() {
        return pubName;
    }
    public void setPubName(String pubName) {
        this.pubName = pubName;
    }
    public int getPrice() {
        return price;
    }
    public void setPrice(int price) {
        this.price = price;
    }
}
```

book

```
package kr.co.bookstore1.vo;

public class CustomerVO {

    private int cusId;
    private String cusName;
    private String cusAddr;
    private String cusHp;

    public int getCusId() {
        return cusId;
    }
    public void setCusId(int cusId) {
        this.cusId = cusId;
    }
    public String getCusName() {
        return cusName;
    }
    public void setCusName(String cusName) {
        this.cusName = cusName;
    }
    public String getCusAddr() {
        return cusAddr;
    }
    public void setCusAddr(String cusAddr) {
        this.cusAddr = cusAddr;
    }
    public String getCusHp() {
        return cusHp;
    }
    public void setCusHp(String cusHp) {
        this.cusHp = cusHp;
    }
}
```

customer

4. 실행화면

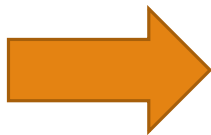
4. 실행화면(index)

← → ↻ ⓘ localhost:8080/BookStore1/

BookStore

[도서 목록](#) [고객 목록](#)

index



도서목록

[처음으로 도서등록](#)

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	수정 삭제
3	민혁이의 일대기1	내머리	9999	수정 삭제

book

← → ↻ ⓘ

localhost:8080/BookStore1/customer/list

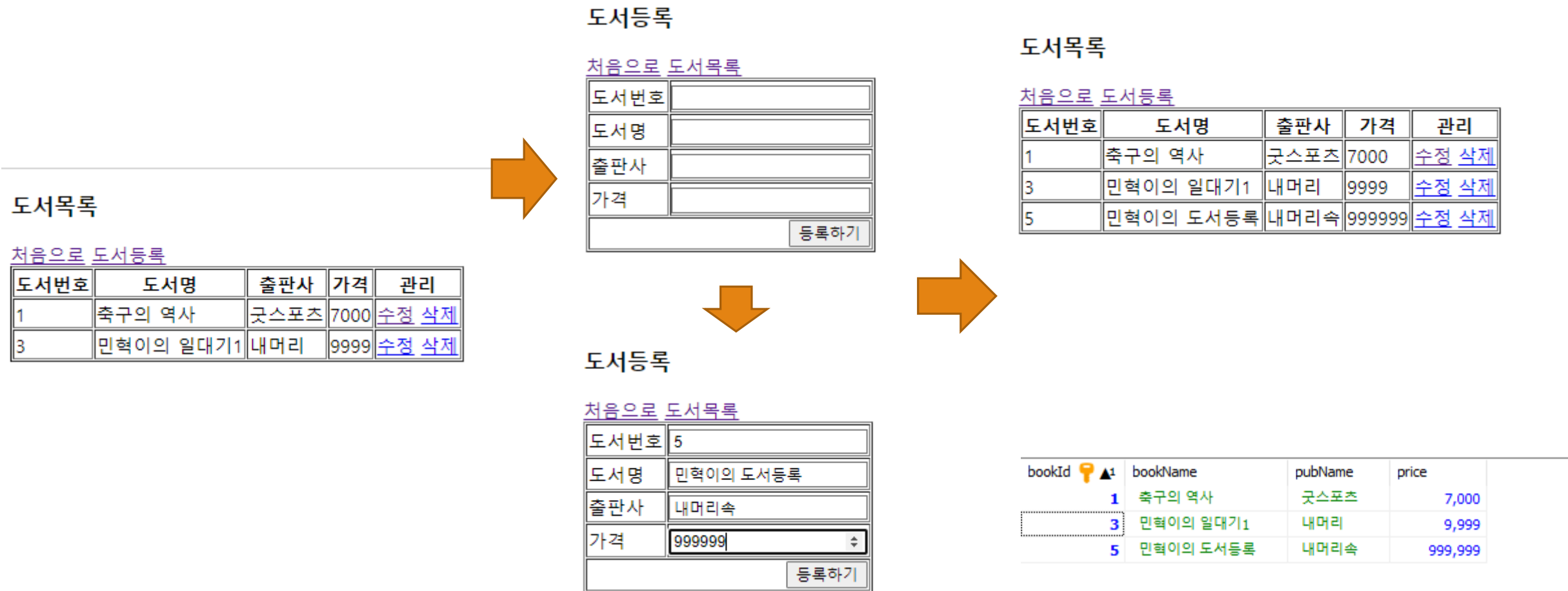
고객목록

[처음으로 고객등록](#)

고객번호	고객명	주소	휴대폰	관리
11	강감찬	부산시 서울구 대구동	010-1234-5678	수정 삭제

customer

4. 실행화면(도서등록)



4. 실행화면(도서수정)

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	수정 삭제
3	민혁이의 일대기1	내머리	9999	수정 삭제
5	민혁이의 도서등록	내머리속	999999	수정 삭제

도서수정

처음으로 도서목록

도서번호	1
도서명	축구의 역사
출판사	굿스포츠
가격	7000
<input type="button" value="수정"/>	

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사(수정)	굿스포츠(수정)	700050	수정 삭제
3	민혁이의 일대기1	내머리	9999	수정 삭제
5	민혁이의 도서등록	내머리속	999999	수정 삭제

도서수정

처음으로 도서목록

도서번호	1
도서명	축구의 역사(수정)
출판사	굿스포츠(수정)
가격	700050
<input type="button" value="수정"/>	

bookId	bookName	pubName	price
1	축구의 역사(수정)	굿스포츠(수정)	700,050
3	민혁이의 일대기1	내머리	9,999
5	민혁이의 도서등록	내머리속	999,999

4. 실행화면(도서삭제)

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사(수정)	굿스포츠(수정)	700050	수정 삭제
3	민혁이의 일대기1	내머리	9999	수정 삭제
5	민혁이의 도서등록	내머리속	999999	수정 삭제



도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
3	민혁이의 일대기1	내머리	9999	수정 삭제
5	민혁이의 도서등록	내머리속	999999	수정 삭제

bookId	bookName	pubName	price
1	축구의 역사(수정)	굿스포츠(수정)	700,050
3	민혁이의 일대기1	내머리	9,999
5	민혁이의 도서등록	내머리속	999,999



bookId	bookName	pubName	price
3	민혁이의 일대기1	내머리	9,999
5	민혁이의 도서등록	내머리속	999,999

4. 실행화면(고객등록)

← → ↻ ⓘ localhost:8080/BookStore1/customer/list

고객목록

[처음으로 고객등록](#)

고객번호	고객명	주소	휴대폰	관리
11	강감찬	부산시 서울구 대구동	010-1234-5678	수정 삭제

고객등록

[처음으로 고객목록](#)

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

고객목록

[처음으로 고객등록](#)

고객번호	고객명	주소	휴대폰	관리
11	강감찬	부산시 서울구 대구동	010-1234-5678	수정 삭제
12	공민혁등록	부산시 금정구 중앙대로	010-5555-5555	수정 삭제

고객등록

[처음으로 고객목록](#)

고객명	공민혁등록
주소	부산시 금정구 중앙대로
휴대폰	010-5555-5555
<input type="button" value="등록하기"/>	

cusId	cusName	cusAddr	cusHp
11	강감찬	부산시 서울구 대구동	010-1234-5678
12	공민혁등록	부산시 금정구 중앙대로	010-5555-5555

4. 실행화면(고객수정)

고객수정

처음으로 고객목록

고객번호	11
고객명	강감찬
주소	부산시 서울구 대구동
휴대폰	010-1234-5678
수정	

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
11	강감찬(수정)	부산시 서울구 대구동(수정)	010-1234-5678	수정 삭제
12	공민혁등록	부산시 금정구 중양대로	010-5555-5555	수정 삭제

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
11	강감찬	부산시 서울구 대구동	010-1234-5678	수정 삭제
12	공민혁등록	부산시 금정구 중양대로	010-5555-5555	수정 삭제

고객수정

처음으로 고객목록

고객번호	11
고객명	강감찬(수정)
주소	부산시 서울구 대구동(수정)
휴대폰	010-1234-5678
수정	

cusId	cusName	cusAddr	cusHp
11	강감찬(수정)	부산시 서울구 대구동(수정)	010-1234-5678
12	공민혁등록	부산시 금정구 중양대로	010-5555-5555

4. 실행화면(고객삭제)

고객목록

[처음으로 고객등록](#)

고객번호	고객명	주소	휴대폰	관리
11	강감찬	부산시 서울구 대구동	010-1234-5678	수정 삭제
12	공민혁등록	부산시 금정구 중앙대로	010-5555-5555	수정 삭제



고객목록

[처음으로 고객등록](#)

고객번호	고객명	주소	휴대폰	관리
12	공민혁등록	부산시 금정구 중앙대로	010-5555-5555	수정 삭제

cusId	cusName	cusAddr	cusHp
11	강감찬(수정)	부산시 서울구 대구동(수정)	010-1234-5678
12	공민혁등록	부산시 금정구 중앙대로	010-5555-5555



cusId	cusName	cusAddr	cusHp
12	공민혁등록	부산시 금정구 중앙대로	010-5555-5555

끝