



#2023.01.04 최아영

수행평가



모던 웹 애플리케이션 개발2



목차

- 스프링 프로그래밍
- RESTful API

#1, 프로젝트 생성 및 구성

#2, 화면 구현

#3, 기능 구현

#4, 실행

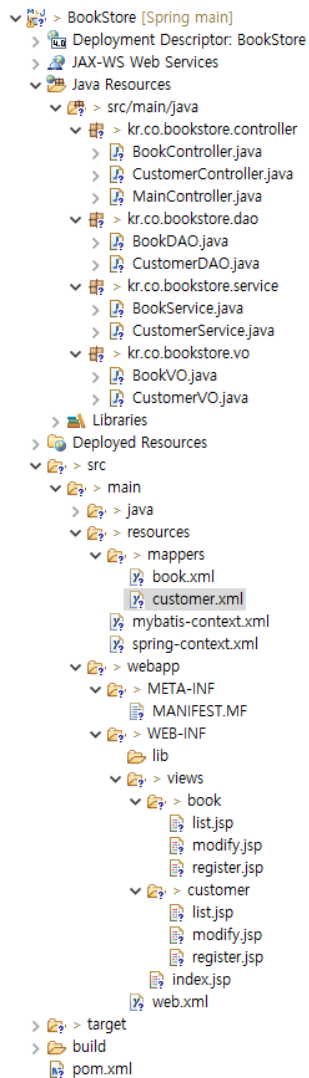


Part 1,

프로젝트 생성 및 구성



프로젝트 생성 및 구성



프로젝트 구조

ProjectName: BookStore

WAS : Tomcat 9

DB : java1_bookstore

Framework: Spring, Mybatis

Tools : Eclipse, Workbench



Part 2,

화면 구현

화면 구현

BookStore

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 단계별 기술	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 Champion	Pearson	13000	수정 삭제

도서수정

처음으로 도서목록

도서번호	10
도서명	Olympic Champion
출판사	Pearson
가격	13000
<input type="button" value="수정"/>	

도서목록 고객목록

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 대전	null	수정 삭제

고객수정

처음으로 고객목록

고객번호	1
고객명	박지성
주소	영국 맨체스터
휴대폰	000-5000-0001
<input type="button" value="수정"/>	

Part 3,

기능 구현



Part 3 기능 구현

[index]

```
<body>
  <h3>BookStore</h3>

  <a href="/BookStore/book/List">도서목록</a>
  <a href="/BookStore/customer/List">고객목록</a>
</body>
```

[pom.xml]

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>kr.co.bookstore</groupId>
  <artifactId>BookStore</artifactId>
  <packaging>war</packaging>
  <version>0.0.1-SNAPSHOT</version>
  <name>Ch05 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>
```

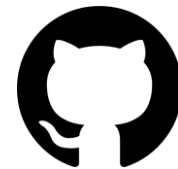
[MainController]

```
@Controller
public class MainController {

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

```
    <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>BookStore</finalName>
  </build>
</project>
```

<코드 GitHub참조>



click

```

<%@ 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="/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 var="book" items="${books}">
        <tr>
          <td>${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>

```

[book-list]

```

<%@ 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="/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 var="cust" items="${custs}">
        <tr>
          <td>${cust.custId}</td>
          <td>${cust.name}</td>
          <td>${cust.address}</td>
          <td>${cust.phone}</td>
          <td>
            <a href="/BookStore/customer/modify?custId=${cust.custId}">수정</a>
            <a href="/BookStore/customer/delete?custId=${cust.custId}">삭제</a>
          </td>
        </tr>
      </c:forEach>
    </table>
  </body>
</html>

```

[customer-list]

```

<body>
<h3>도서수정</h3>
<a href="/BookStore/book/list">도서목록</a>
<form action="/BookStore/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="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>

```

[book-modify]

```

<body>
<h3>고객수정</h3>
<a href="/BookStore/customer/list">고객목록</a>
<form action="/BookStore/customer/modify" method="post">
  <table border="1">
    <tr>
      <td>고객번호</td>
      <td><input type="text" name="custId" readonly value="${cust.custId}"/></td>
    </tr>
    <tr>
      <td>고객명</td>
      <td><input type="text" name="name" value="${cust.name}"/></td>
    </tr>
    <tr>
      <td>주소</td>
      <td><input type="text" name="address" value="${cust.address}"/></td>
    </tr>
    <tr>
      <td>휴대폰</td>
      <td><input type="text" name="phone" value="${cust.phone}"/></td>
    </tr>
    <tr>
      <td colspan="2" align="right">
        <input type="submit" value="수정"/>
      </td>
    </tr>
  </table>
</form>
</body>

```

[customer-modify]

```

<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>도서번호</td>
        <td><input type="text" name="bookId" placeholder="도서번호 입력"/></td>
      </tr>
      <tr>
        <td>도서명</td>
        <td><input type="text" name="bookName" placeholder="도서명 입력"/></td>
      </tr>
      <tr>
        <td>출판사</td>
        <td><input type="text" name="publisher" placeholder="출판사 입력"/></td>
      </tr>
      <tr>
        <td>가격</td>
        <td><input type="number" name="price" placeholder="가격 입력"/></td>
      </tr>
      <tr>
        <td colspan="2" align="right">
          <input type="submit" value="등록"/>
        </td>
      </tr>
    </table>
  </form>
</body>

```

[book-register]

```

<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" placeholder="고객명 입력"/></td>
      </tr>
      <tr>
        <td>주소</td>
        <td><input type="text" name="address" placeholder="주소 입력"/></td>
      </tr>
      <tr>
        <td>휴대폰</td>
        <td><input type="text" name="phone" placeholder="휴대폰 입력"/></td>
      </tr>
      <tr>
        <td colspan="2" align="right">
          <input type="submit" value="등록"/>
        </td>
      </tr>
    </table>
  </form>
</body>

```

[customer-register]


```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://xmlns.jcp.org/"
  <display-name>BookStore</display-name>
  <welcome-file-list>
    <welcome-file>index.html</welcome-file>
    <welcome-file>index.jsp</welcome-file>
    <welcome-file>index.htm</welcome-file>
    <welcome-file>default.html</welcome-file>
    <welcome-file>default.jsp</welcome-file>
    <welcome-file>default.htm</welcome-file>
  </welcome-file-list>
  <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>
  <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>
</web-app>
```

```
public class BookVO {
    private int bookId;
    private String bookName;
    private String publisher;
    private String 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 getPublisher() {
        return publisher;
    }
    public void setPublisher(String publisher) {
        this.publisher = publisher;
    }
    public String getPrice() {
        return price;
    }
    public void setPrice(String price) {
        this.price = price;
    }
}
```

[BookVO]

```
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 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;
    }
}
```

[CustomerVO]

```
@Controller
public class BookController {

    @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 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";
}

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

    service.deleteBook(bookId);

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

Part 3 기능 구현 [kr.co.bookstore.controller - CustomerController]

```
@Controller
public class CustomerController {

    @Autowired
    private CustomerService service;

    @GetMapping("/customer/list")
    public String list(Model model) {
        List<CustomerVO> custs = service.selectCustomers();
        model.addAttribute("custs", custs);
        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 custId) {
        CustomerVO cust = service.selectCustomer(custId);
        model.addAttribute("cust", cust);
        return "/customer/modify";
    }

    @PostMapping("/customer/modify")
    public String modify(CustomerVO vo) {
        service.updateUser1(vo);
        return "redirect:/customer/list";
    }

    @GetMapping("/customer/delete")
    public String delete(String custId) {
        service.deleteCustomer(custId);
        return "redirect:/customer/list";
    }
}
```


기능 구현 [resources]

```
<?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.bookstore"/>

    <!-- 스프링 Web MVC 관련 어노테이션을 위한 태그선언 -->
    <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 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>
```

[spring-context.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>
```

[mybatis-context.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">
<mapper namespace="book">
  <insert id="insertBook">
    INSERT INTO `book` VALUES (#{bookId}, #{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>
```

[book.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">
<mapper namespace="customer">
  <insert id="insertCustomer">
    INSERT INTO `customer` VALUES (#{custId}, #{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>
```

[customer.xml]

A close-up photograph of a brown leather bag with a strap, resting on a dark blue textured surface. Inside the bag, a black smartphone, a brown notebook with a red string, a red and silver pen, and a pair of black-rimmed glasses are visible.

Part 4,

실행

BookStore

도서목록 고객목록

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	수정 삭제
2	축구아는 여자	나무수	13000	수정 삭제
3	축구의 이해	대한미디어	22000	수정 삭제
4	골프 바이블	대한미디어	35000	수정 삭제
5	피겨 교본	굿스포츠	8000	수정 삭제
6	역도 단계별 기술	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 Champion	Pearson	13000	수정 삭제

도서수정

처음으로 도서목록

도서번호	1
도서명	축구의 역사
출판사	굿스포츠
가격	7000
<div>수정</div>	

도서수정

처음으로 도서목록

도서번호	1
도서명	축구의 역사
출판사	굿스포츠1
가격	7100
<div>수정</div>	

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠1	7100	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 단계별 기술	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 Champion	Pearson	13000	수정 삭제

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
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 단계별 기술	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 Champion	Pearson	13000	수정 삭제

BookStore

[도서목록](#) [고객목록](#)

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 대전	null	수정 삭제

고객수정

처음으로 고객목록

고객번호	1
고객명	박지성
주소	영국 맨체스터
휴대폰	000-5000-0001
<input type="button" value="수정"/>	

고객수정

처음으로 고객목록

고객번호	1
고객명	박지성
주소	영국 런던
휴대폰	000-5000-0002
<input type="button" value="수정"/>	

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박지성	영국 런던	000-5000-0002	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 대전	null	수정 삭제

고객목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
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 대전	null	수정 삭제

감사합니다.