

서버프로그램 구현

서정현

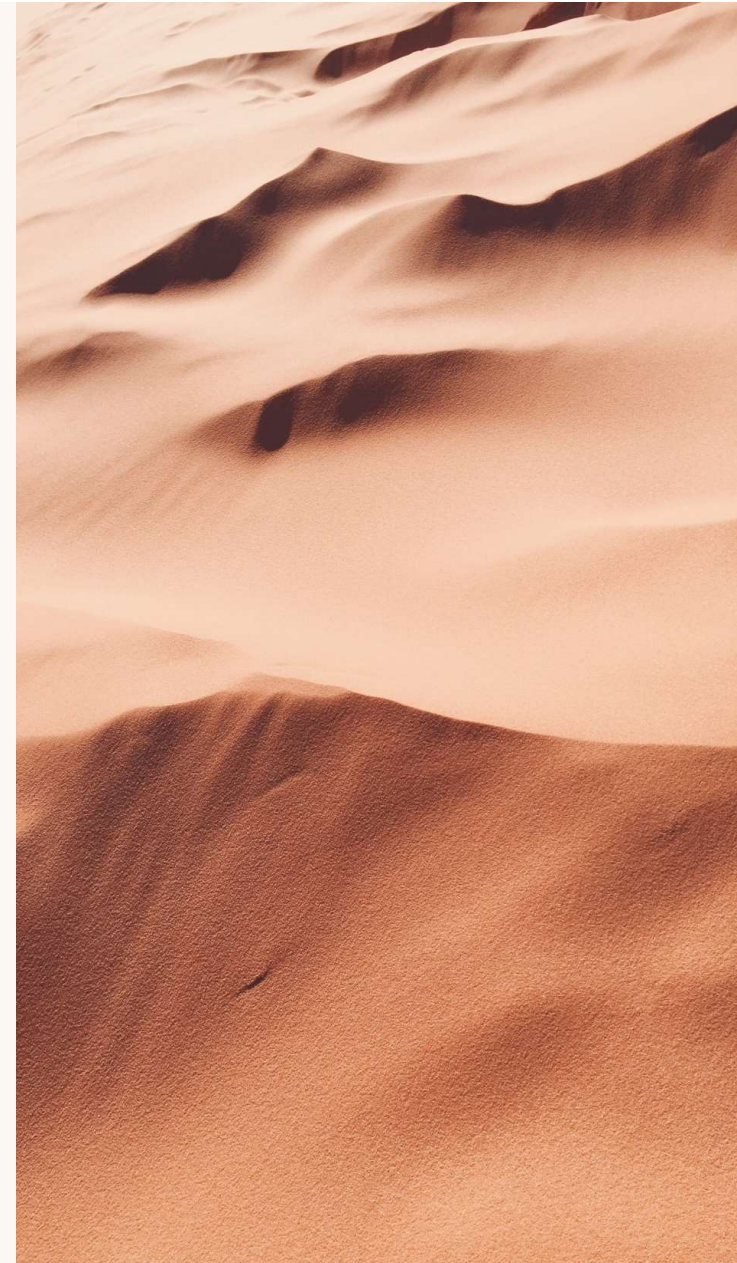
목차 A table of contents.

1.문제 1 - 프로젝트 생성 및 구성

2.문제 2 - 화면 구현

3.문제 3 - 기능 구현

4.문제 4 - 실행



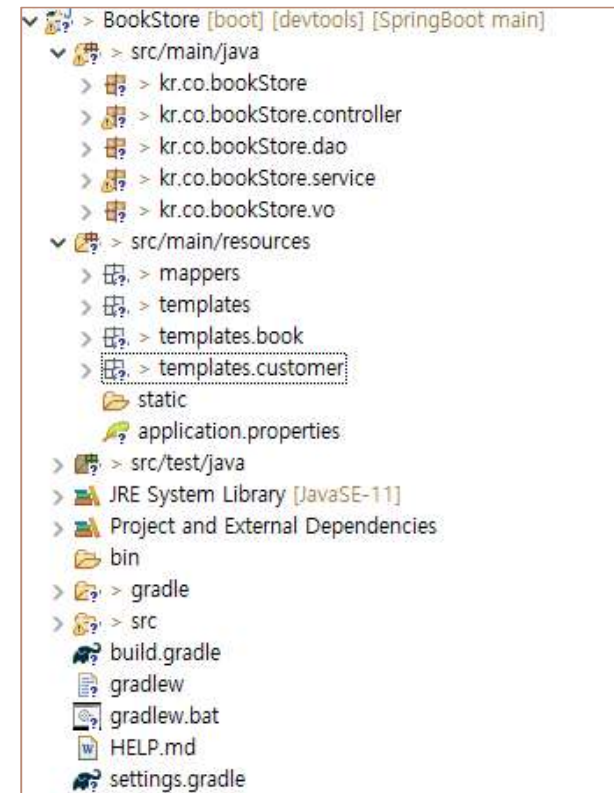
1. 문제 1 – 프로젝트 생성 및 구성

1. 프로젝트명 : BookStore

2. 데이터베이스 : java2_bookstore

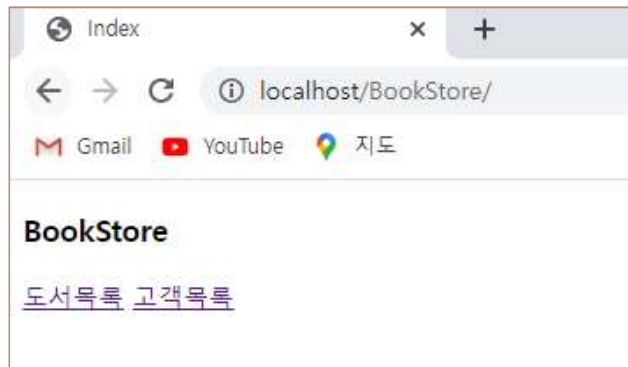
3. 개발 도구

- WAS : Tomcat9
- Framework : Spring(SpringBoot), Mybatis
- Tools : Eclipse, Workbench



2. 문제 2 – 화면 구현(메인)

1. 메인 화면



2. 문제 2 – 화면 구현(도서)

1. 도서 목록

book::list

localhost/BookStore/book/list

도서 목록

[처음으로 도서등록](#)

도서번호	도서명	출판사	가격	관리
1	축구의 역사?	굿스포츠	8000	수정 삭제
2	축구하는 여자	나무수	13000	수정 삭제
3	축구의 이해	대한미디어	22000	수정 삭제
4	골프 바이블	대한미디어	35000	수정 삭제
5	피겨 교본	굿스포츠	8000	수정 삭제
6	역도 단계별기술	굿스포츠	6000	수정 삭제
7	야구의 추억	이상미디어	20000	수정 삭제
8	야구를 부탁해	이상미디어	13000	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제
11	소나기2	동화	2000	수정 삭제
102	소나무	동화나라	5000	수정 삭제
103	상상의 나라	무궁화	5300	수정 삭제
105	asdas111	asdasd	2000	수정 삭제

2. 도서 등록

user1::register

localhost/BookStore/book/register

도서 등록

[처음으로 도서 목록](#)

도서명	<input type="text"/>
출판사	<input type="text"/>
가격	<input type="text"/>
<input type="button" value="등록"/>	

3. 도서 수정

localhost/BookStore/book/modify?bookId=1

도서 수정

[처음으로 도서목록](#)

도서번호	<input type="text" value="1"/>
도서명	<input type="text" value="축구의 역사?"/>
출판사	<input type="text" value="굿스포츠"/>
가격	<input type="text" value="8000"/>
<input type="button" value="수정"/>	

2. 문제 2 – 화면 구현(고객)

1. 고객 목록

book::list

localhost/BookStore/customer/list

고객 목록

[처음으로 고객등록](#)

고객번호	고객명	주소	휴대폰	관리
1	박지성2	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 클리블랜드2	000-8000-0001	수정 삭제
6	홍길동	제주도	010-1234-1111	수정 삭제
7	김구2	부산광역시 부산진구	010-1234-2222	수정 삭제

2. 고객 등록

user1::register

localhost/BookStore/customer/register

고객 등록

[처음으로 고객 목록](#)

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

3. 고객 수정

user1::modify

localhost/BookStore/customer/modify?custId=1

고객 수정

[처음으로 고객 목록](#)

고객번호	<input type="text" value="1"/>
고객명	<input type="text" value="박지성2"/>
주소	<input type="text" value="영국 맨체스타"/>
휴대폰	<input type="text" value="000-5000-0001"/>
<input type="button" value="수정"/>	

3. 문제 3 – 기능 구현(도서)

1. Controller

```
@Controller
@RequestMapping("/book")
public class BookController {

    @Autowired
    private BookService service;

    @GetMapping("/list")
    public void list(Model model) {
        model.addAttribute("books", service.selectBooks());
    }

    @GetMapping("/register")
    public void register() {}

    @PostMapping("/register")
    public void register(BookVO vo) {
        service.insertBook(vo);
    }

    @GetMapping("/modify")
    public void modify(Model model, String bookId) {
        model.addAttribute("book", service.selectBook(bookId));
    }

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

    @GetMapping("/delete")
    public String delete(String bookId) {
        service.deleteBook(bookId);
        return "redirect:/book/list";
    }
}
```

2. Service

```
@Service
public class BookService {

    @Autowired
    private BookDAO dao;

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

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

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

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

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

3. Dao

```
@Repository
public class BookDAO {

    @Autowired
    SqlSessionTemplate mybatis;

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

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

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

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

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


3. 문제 3 – 기능 구현(도서)

4. VO

```
import lombok.AllArgsConstructor;

@Setter
@Getter
@AllArgsConstructor
@NoArgsConstructor
@ToString
public class BookVO {
    private int bookId;
    private String bookname;
    private String publisher;
    private int price;
}
```

2. 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="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>
  <delete id="updateBook">
    UPDATE `book` SET
      `bookname`=#{bookname},
      `publisher`=#{publisher},
      `price`=#{price}
    WHERE
      `bookId`=#{bookId};
  </delete>
  <update id="deleteBook">
    DELETE FROM `book` WHERE `bookId`=#{bookId};
  </update>
</mapper>
```


3. 문제 3 – 기능 구현(고객)

1. Controller

```
@Controller
@RequestMapping("/customer")
public class CustomerController {

    @Autowired
    private CustomerService service;

    @GetMapping("/list")
    public void list(Model model) {
        model.addAttribute("customers", service.selectCustomers());
    }

    @GetMapping("/register")
    public void register() {}

    @PostMapping("/register")
    public void register(CustomerVO vo) {
        service.insertCustomer(vo);
    }

    @GetMapping("/modify")
    public void modify(Model model, String custId) {
        model.addAttribute("customer", service.selectCustomer(custId));
    }

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

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

2. Service

```
@Service
public class CustomerService {

    @Autowired
    private CustomerDAO dao;

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

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

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

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

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

3. Dao

```
@Repository
public class CustomerDAO {

    @Autowired
    SqlSessionTemplate mybatis;

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

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

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

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

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

3. 문제 3 – 기능 구현(고객)

4. VO

```
@Setter
@Getter
@AllArgsConstructor
@NoArgsConstructor
@ToString
public class CustomerVO {
    private int custId;
    private String name;
    private String address;
    private String phone;
}
```

2. customer.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` (`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>
    <delete id="updateCustomer">
        UPDATE `customer` SET
            `name`=#{name},
            `address`=#{address},
            `phone`=#{phone}
        WHERE
            `custId`=#{custId};
    </delete>
    <update id="deleteCustomer">
        DELETE FROM `customer` WHERE `custId`=#{custId};
    </update>
</mapper>
```

4. 문제 4 – 실행(도서)

1. 도서 목록 출력 문제1번 화면 참고

2. 도서 등록

도서 등록

처음으로 도서 목록

도서명

이습우화

출판사

동화나라

가격

10000

등록

도서 목록

처음으로 도서목록

도서번호	도서명	출판사	가격	관리
1	축구의 역사?	굿스포즈	8000	수정 삭제
2	축구는 여자	나무수	13000	수정 삭제
3	축구의 이해	대한미디어	22000	수정 삭제
4	골프 바이블	대한미디어	35000	수정 삭제
5	피겨 교본	굿스포즈	8000	수정 삭제
6	역도 단계별기술	굿스포즈	6000	수정 삭제
7	야구의 추억	이상미디어	20000	수정 삭제
8	야구를 부탁해	이상미디어	13000	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제
11	소나기2	동화	2000	수정 삭제
102	소나무	동화나라	5000	수정 삭제
103	상상의 나라	무궁화	5300	수정 삭제
105	asdas111	asdasd	2000	수정 삭제
106	이습우화	동화나라	10000	수정 삭제

3. 도서 수정

도서 수정

처음으로 도서목록

도서번호

106

도서명

이습우화1

출판사

동화나라

가격

10000

수정

도서 목록

처음으로 도서목록

도서번호	도서명	출판사	가격	관리
1	축구의 역사?	굿스포즈	8000	수정 삭제
2	축구는 여자	나무수	13000	수정 삭제
3	축구의 이해	대한미디어	22000	수정 삭제
4	골프 바이블	대한미디어	35000	수정 삭제
5	피겨 교본	굿스포즈	8000	수정 삭제
6	역도 단계별기술	굿스포즈	6000	수정 삭제
7	야구의 추억	이상미디어	20000	수정 삭제
8	야구를 부탁해	이상미디어	13000	수정 삭제
10	Olympic Champions	Pearson	13000	수정 삭제
11	소나기2	동화	2000	수정 삭제
102	소나무	동화나라	5000	수정 삭제
103	상상의 나라	무궁화	5300	수정 삭제
105	asdas111	asdasd	2000	수정 삭제
106	이습우화1	동화나라	10000	수정 삭제

4. 문제 4 – 실행(고객)

1. 고객 목록 출력 문제1번 화면 참고
2. 고객 등록

고객 등록

처음으로 고객 목록

고객명	아무개
주소	미국 클리블랜드
휴대폰	010-1234-2222
<input type="button" value="등록"/>	



고객 목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박지성2	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 클리블랜드2	000-8000-0001	수정 삭제
6	홍길동	제주도	010-1234-1111	수정 삭제
7	김구2	부산광역시 부산진구	010-1234-2222	수정 삭제
12	아무개	미국 클리블랜드	010-1234-2222	수정 삭제

3. 고객 수정

고객 수정

처음으로 고객목록

고객번호	12
고객명	아무개1
주소	미국 클리블랜드
휴대폰	010-1234-2222
<input type="button" value="수정"/>	



고객 목록

처음으로 고객등록

고객번호	고객명	주소	휴대폰	관리
1	박지성2	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 클리블랜드2	000-8000-0001	수정 삭제
6	홍길동	제주도	010-1234-1111	수정 삭제
7	김구2	부산광역시 부산진구	010-1234-2222	수정 삭제
12	아무개1	미국 클리블랜드	010-1234-2222	수정 삭제

감사합니다