



프로그래밍 기초 및 응용

서버 프로그램 구현





Contents

1

문제 1. 프로그램 생성 및 구성

2

문제 2. 화면 구현

3

문제 3. 기능 구현

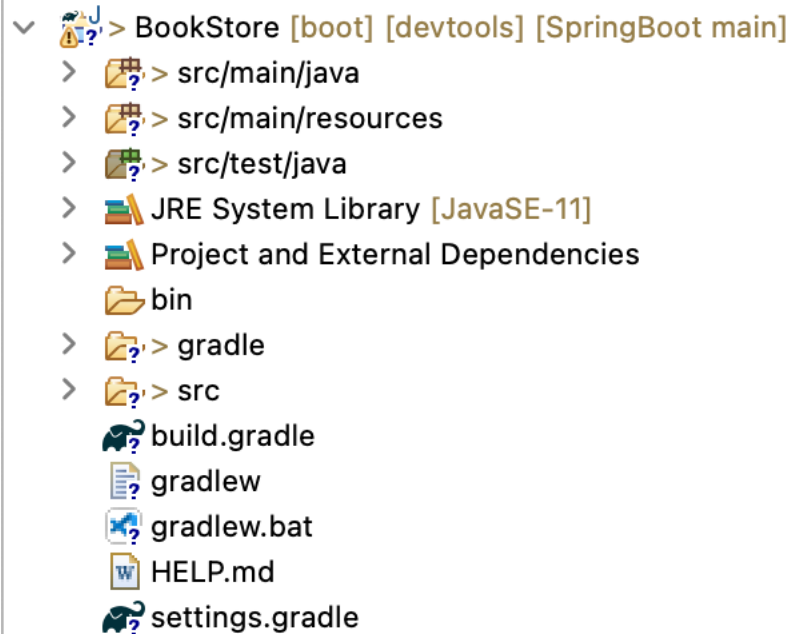
4

문제 4. 실행

1. 프로그램 생성 및 구성

大

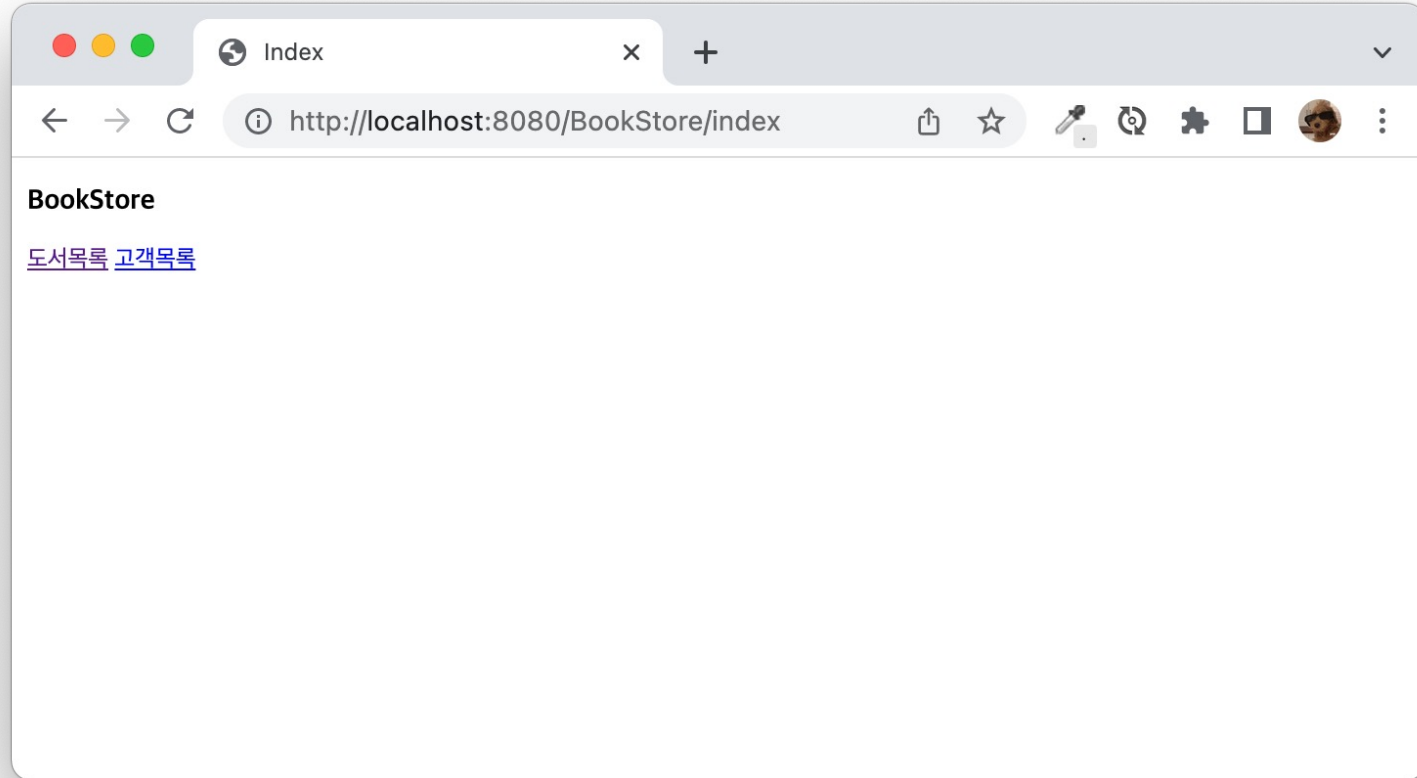
- ProjectName: BookStore
- WAS: Tomcat9
- DB: java2_BookStore
- Framework: Spring(SpringBoot), Bybatis
- Tools: Eclipse, Workbench



2. 화면 구현

大

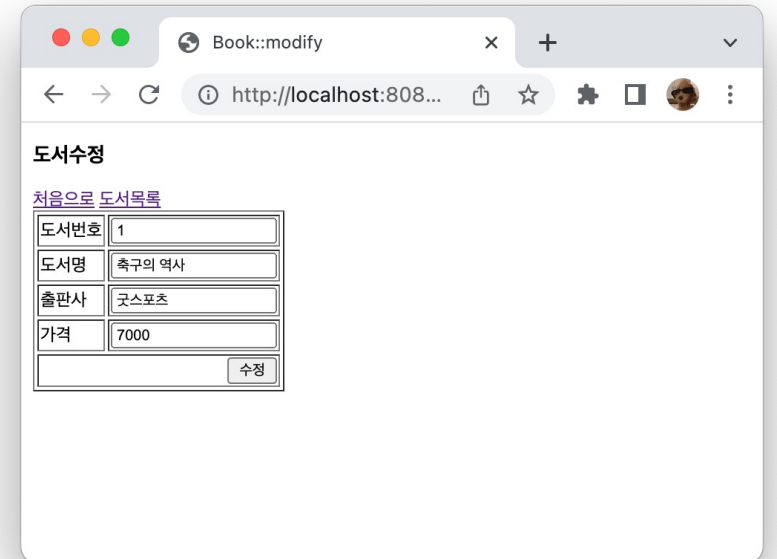
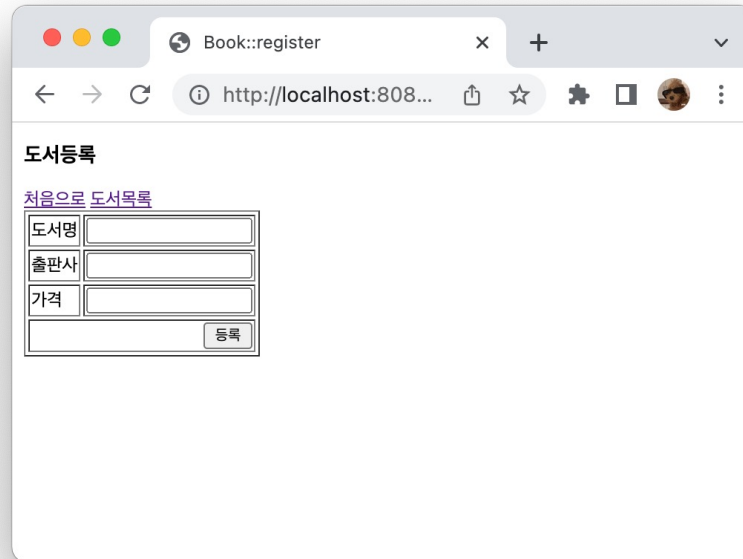
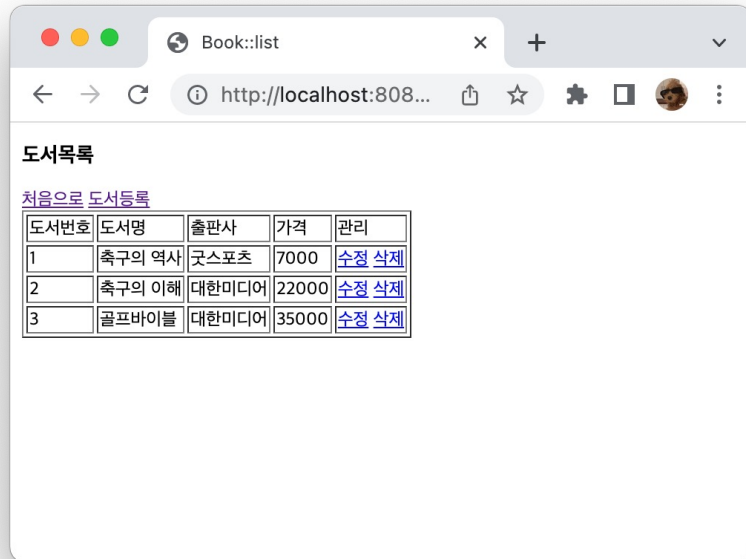
(1) index



2. 화면 구현

大

(2) 도서 목록, 등록, 수정 화면 구현



2. 화면 구현

大

(3) 고객 목록, 등록, 수정 화면 구현

Customer::list

http://localhost:...

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
5	박세리	대한민국 대전	010-1112-1123	수정 삭제
6	이강철	제주특별자치시	010-9999-9999	수정 삭제

Customer::register

http://localhost:...

고객등록

[처음으로 고객목록](#)

고객번호

고객명

주소

휴대폰

수정

Customer::modify

http://localhost:...

고객수정

[처음으로 고객목록](#)

고객번호

1

고객명

박지성

주소

영국 맨체스타

휴대폰

000-5000-0001

수정

3. 기능 구현 - 도서

大

Controller

```
@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(String bookId, Model model) {
        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";
    }
}
```

Service

```
@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);
    }
}
```

DAO

```
@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);
    }
}
```

VO

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

3. 기능 구현 - 고객

大

Controller

```
@Controller
public class CustomerController {

    @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(String custId, Model model) {
        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";
    }

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

Service

```
@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);
    }
}
```

VO

```
@Getter
@Setter
@AllArgsConstructor
@NoArgsConstructor
@ToString
public class CustomerVO {

    private int custId;
    private String name;
    private String address;
    private String phone;
}
```

DAO

```
@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);
    }
}
```


4. 실행

大

(1) 도서 목록 및 등록

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	수정 삭제
2	축구의 이해	대한미디어	22000	수정 삭제
3	골프바이블	대한미디어	35000	수정 삭제

도서등록

처음으로 도서목록

도서명	물고기는 존재하지 않는다
출판사	곰출판
가격	15300
<div>등록</div>	

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	수정 삭제
2	축구의 이해	대한미디어	22000	수정 삭제
3	골프바이블	대한미디어	35000	수정 삭제
4	물고기는 존재하지 않는다	곰출판	15300	수정 삭제

(2) 도서 수정

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	수정 삭제
2	축구의 이해	대한미디어	22000	수정 삭제
3	골프바이블	대한미디어	35000	수정 삭제
4	물고기는 존재하지 않는다	곰출판	15300	수정 삭제

도서수정

처음으로 도서목록

도서번호	4
도서명	물고기는 존재하지 않는다
출판사	곰출판
가격	10000
<div>수정</div>	

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	수정 삭제
2	축구의 이해	대한미디어	22000	수정 삭제
3	골프바이블	대한미디어	35000	수정 삭제
4	물고기는 존재하지 않는다	곰출판	10000	수정 삭제

4. 실행

大

(3) 도서 삭제

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	수정 삭제
2	축구의 이해	대한미디어	22000	수정 삭제
3	골프바이블	대한미디어	35000	수정 삭제
4	물고기는 존재하지 않는다	곰출판	10000	수정 삭제

도서목록

처음으로 도서등록

도서번호	도서명	출판사	가격	관리
1	축구의 역사	굿스포츠	7000	수정 삭제
2	축구의 이해	대한미디어	22000	수정 삭제
3	골프바이블	대한미디어	35000	수정 삭제

4. 실행

(1) 고객 목록 및 등록

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
5	박세리	대한민국 대전	010-1112-1123	수정 삭제
6	이감굴	제주특별자치시	010-9999-9999	수정 삭제

고객등록

처음으로 고객목록

고객번호	7
고객명	박코딩
주소	부산시 동래구
휴대폰	010-5555-0000
<div>등록</div>	

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
5	박세리	대한민국 대전	010-1112-1123	수정 삭제
6	이감굴	제주특별자치시	010-9999-9999	수정 삭제
7	박코딩	부산시 동래구	010-5555-0000	수정 삭제

(2) 고객 수정

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
5	박세리	대한민국 대전	010-1112-1123	수정 삭제
6	이감굴	제주특별자치시	010-9999-9999	수정 삭제
7	박코딩	부산시 동래구	010-1111-2222	수정 삭제

고객수정

처음으로 고객목록

고객번호	7
고객명	박코딩
주소	부산시 동래구
휴대폰	010-1111-2222
<div>수정</div>	

4. 실행

大

(3) 고객 삭제

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
5	박세리	대한민국 대전	010-1112-1123	수정 삭제
6	이감굴	제주특별자치시	010-9999-9999	수정 삭제
7	박코딩	부산시 동래구	010-1111-2222	수정 삭제

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
5	박세리	대한민국 대전	010-1112-1123	수정 삭제
6	이감굴	제주특별자치시	010-9999-9999	수정 삭제