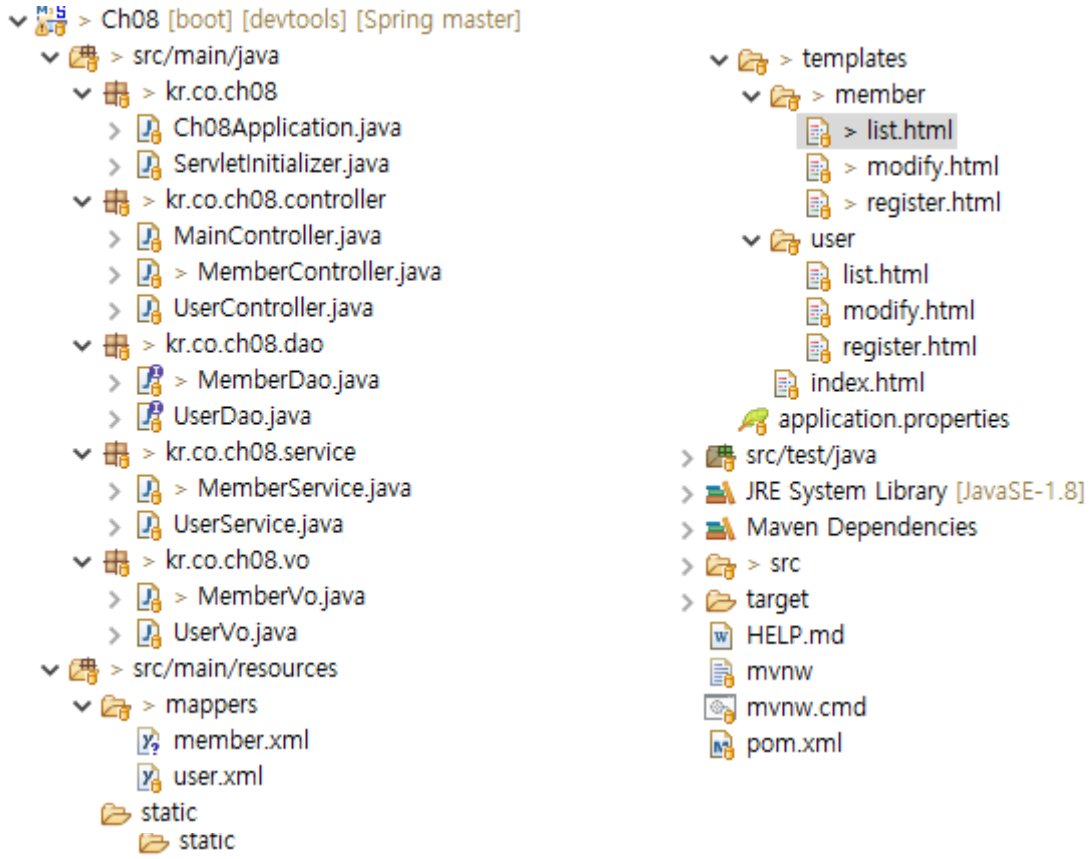
통합구현

김 나 현

프로젝트 구조


Ch08 [boot] [devtools] [Spring master]
src/main/java
kr.co.ch08
Ch08Application.java
ServletInitializer.java
kr.co.ch08.controller
MainController.java
MemberController.java
UserController.java
kr.co.ch08.dao
MemberDao.java
UserDao.java
kr.co.ch08.service
MemberService.java
UserService.java
kr.co.ch08.vo
MemberVo.java
UserVo.java
src/main/resources
mappers
member.xml
user.xml
static
static
templates
member
list.html
modify.html
register.html
user
list.html
modify.html
register.html
index.html
application.properties
src/test/java
JRE System Library [JavaSE-1.8]
Maven Dependencies
src
target
HELP.md
mvnw
mvnw.cmd
pom.xml

pom.xml

```xml
<dependency>
    <groupId>org.mybatis.spring.boot</groupId>
    <artifactId>mybatis-spring-boot-starter</artifactId>
    <version>2.1.4</version>
</dependency>
```

Spring boot에 mybatis를 사용하기위해
환경 설정

Ch08Application.java

```java
package kr.co.ch08;

import org.mybatis.spring.annotation.MapperScan;

@SpringBootApplication
@MapperScan("kr.co.ch08.dao")
public class Ch08Application {

    public static void main(String[] args) {
        SpringApplication.run(Ch08Application.class, args);
    }

}
```

mapper 파일을 사용하기 위해
@MapperScan 어노테이션 설정.

application.properties

```
#livereload Setting(Must install Chrome livereload plugin First)
spring.devtools.livereload.enabled=true
spring.thymeleaf.cache=false

#Context root setting
server.servlet.context-path=/Ch08

#DB Setting
spring.datasource.url=jdbc:mysql://192.168.10.114:3306/knh
spring.datasource.username=knh
spring.datasource.password=1234
spring.datasource.driver-class-name=com.mysql.jdbc.Driver

#Mybatis Setting(Ch08 Application add @MapperScan(kr.co.ch08.dao))
# mappers/**/*.xml -> mappers folder below...all..xml..file...
mybatis.mapper-locations=classpath:mappers/**/*.xml

```

경로 지정 및
DB 연결을 위한 mybatis ORM 사용 설정

MemberVo.java

```java
package kr.co.ch08.vo;

import lombok.Getter;

@Getter
@Setter
public class MemberVo {

    private String uid;
    private String name;
    private String hp;
    private String pos;
    private int dep;
    private String rdate;

}
```

DB의 컬럼값을 받아줄 MemberVo.
@Getter @Setter 어노테이션으로 자동으로
get, set 메소드 설정

member.xml

```xml
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN" "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="kr.co.ch08.dao.MemberDao">

    <insert id="insertMember">
        INSERT INTO `MEMBER` VALUES (#{uid}, #{name}, #{hp}, #{pos}, #{dep}, NOW());
    </insert>

    <select id="selectMember" resultType="kr.co.ch08.vo.MemberVo">
        SELECT * FROM `MEMBER` WHERE `uid`=#{uid};
    </select>
    <select id="selectMembers" resultType="kr.co.ch08.vo.MemberVo">
        SELECT * FROM `MEMBER`;
    </select>
    <update id="updateMember">
        UPDATE `MEMBER` SET
                        `name`=#{name},
                        `hp`=#{hp},
                        `pos`=#{pos},
                        `dep`=#{dep}
                       WHERE
                        `uid`=#{uid};
    </update>
    <delete id="deleteMember">
        DELETE FROM `MEMBER` WHERE `uid`=#{uid};
    </delete>

</mapper>
```

MemberDao를 mapper 파일과 연결

member.xml에 시스템에서 사용할 쿼리문 저장.

MemberDao.java

```java
package kr.co.ch08.dao;

import java.util.List;

import org.springframework.stereotype.Repository;

import kr.co.ch08.vo.MemberVo;

@Repository
public interface MemberDao {

    public void insertMember(MemberVo vo);
    public MemberVo selectMember(String uid);
    public List<MemberVo> selectMembers();
    public void updateMember(MemberVo vo);
    public void deleteMember(String uid);

}
```

@Repository 어노테이션으로
member.xml에 저장되어 있는 쿼리문을 사용하여
데이터 불러오기

MemberService.java

```java
package kr.co.ch08.service;

import java.util.List;

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

import kr.co.ch08.dao.MemberDao;
import kr.co.ch08.vo.MemberVo;

@Service
public class MemberService {

    @Autowired
    private MemberDao dao;

    public void insertMember(MemberVo vo) {
        dao.insertMember(vo);
    };
    public MemberVo selectMember(String uid) {
        return dao.selectMember(uid);
    };
    public List<MemberVo> selectMembers() {
        return dao.selectMembers();
    };
    public void updateMember(MemberVo vo) {
        dao.updateMember(vo);
    };
    public void deleteMember(String uid) {
        dao.deleteMember(uid);
    };
}
```

MemberDao의 메소드를 호출.
Controller에서 최종적으로 사용하기 위해
@Service 어노테이션 설정.

MemberController.java

```java
package kr.co.ch08.controller;

import java.util.List;

@Controller
public class MemberController {

    @Autowired
    private MemberService service;

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

        List<MemberVo> members = service.selectMembers();

        model.addAttribute("members", members);

        return "/member/list";

    }

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

    @PostMapping("/member/register")
    public String register(MemberVo vo) {
        service.insertMember(vo);
        return "redirect:/member/list";
    }
```

```java
@GetMapping("/member/modify")
public String modify(String uid, Model model) {

    MemberVo member = service.selectMember(uid);

    model.addAttribute(member);

    return "/member/modify";
}

@PostMapping("/member/modify")
public String modify(MemberVo vo) {

    service.updateMember(vo);

    return "redirect:/member/list";
}


@GetMapping("/member/delete")
public String delete(String uid) {

    service.deleteMember(uid);

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

쿼리문(메소드) 사용을 위해 @Autowired 어노테이션 사용하여
MemberService의 service 객체 생성.
@Controller 어노테이션을 이용하여Client가 요청한 페이지에 대한
작업을 진행.

Post로 데이터가 들어오는 페이지는
@PostMapping 어노테이션 사용.
나머지는 @GetMapping 사용.

Index.html

스프링부트 MyBatis 실습하기

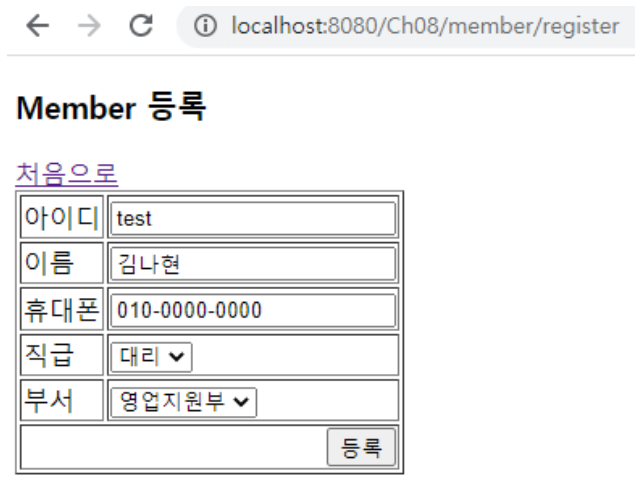
User 등록 User 목록 Member 등록 Member 목록

구현 화면

```html
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>index</title>
</head>
<body>
    <h1>스프링부트 MyBatis 실습하기</h1>

    <a th:href="@{/user/register}">User 등록</a>
    <a th:href="@{/user/list}">User 목록</a>

    <a th:href="@{/member/register}">Member 등록</a>
    <a th:href="@{/member/list}">Member 목록</a>

</body>
</html>
```

코드

register.html



등록할 데이터 입력

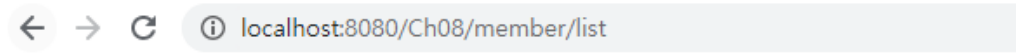
```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>register</title>
</head>
<body>
    <h3>Member 등록</h3>
    <a th:href="@{/}">처음으로</a>
    <form th:action="@{/member/register}" method="post">
        <table border="1">
            <tr>
                <td>아이디</td>
                <td><input type="text" name="uid"/></td>
            </tr>
            <tr>
                <td>이름</td>
                <td><input type="text" name="name"/></td>
            </tr>
            <tr>
                <td>휴대폰</td>
                <td><input type="text" name="hp"/></td>
            </tr>
            <tr>
                <td>직급</td>
                <td>
                    <select name="pos">
                        <option>사원</optio
                        <option>대리</optio
                        <option>과장</optio
                        <option>차장</optio
                        <option>부장</optio
                    </select>
                </td>
            </tr>
            <tr>
                <td>부서</td>
                <td>
                    <select name="dep">
                        <option value="101">영업1부</option>
                        <option value="102">영업2부</option>
                        <option value="103">영업3부</option>
                        <option value="104">영업지원부</option>
                        <option value="105">인사부</option>
                    </select>
                </td>
            </tr>
            <tr>
                <td colspan="2" align="right"><input type="submit" value="등록"/></td>
            </tr>
        </table>
    </form>
</body>
</html>
```

register에 post로 정보가 전송되면 controller에 의해 service.insertMember(vo) 실행으로 데이터 삽입 후, list로 redirect

구현 화면

코드

list.html



Member 목록

Register.html에서 member를 등록한 결과 list

처음으로

아이디	이름	휴대폰	직급	부서	입사일	관리
a101	박혁거세	010-1234-1001	부장	101	2021-03-25 14:16:57	수정 삭제
a102	김유신	010-1234-1002	차장	101	2021-03-24 15:43:46	수정 삭제
a103	김춘추	010-1234-1003	사원	101	2021-03-24 15:43:46	수정 삭제
a104	장보고	010-1234-1004	대리	102	2021-03-24 15:43:46	수정 삭제
a105	강감찬	010-1234-1005	과장	102	2021-03-24 15:43:46	수정 삭제
a106	이성계	010-1234-1006	차장	103	2021-03-24 15:43:46	수정 삭제
a107	정철	010-1234-1007	차장	103	2021-03-24 15:43:46	수정 삭제
a108	이순신	010-1234-1008	부장	104	2021-03-24 15:43:46	수정 삭제
a109	허균	010-1234-1009	부장	104	2021-03-24 15:43:46	수정 삭제
a110	정약용	010-1234-1010	사원	105	2021-03-24 15:43:46	수정 삭제
a111	박지원	010-1234-1011	사원	105	2021-03-24 15:43:46	수정 삭제
b101	을지문덕	010-5555-1234	차장	107	2021-03-24 16:16:14	수정 삭제
test	김나현	010-0000-0000	대리	104	2021-04-23 12:28:28	수정 삭제

대리에서 과장으로 승진 예정

수정 버튼을 누르면 해당 데이터의 uid를 가지고 데이터가 modify.html 화면으로 넘어감

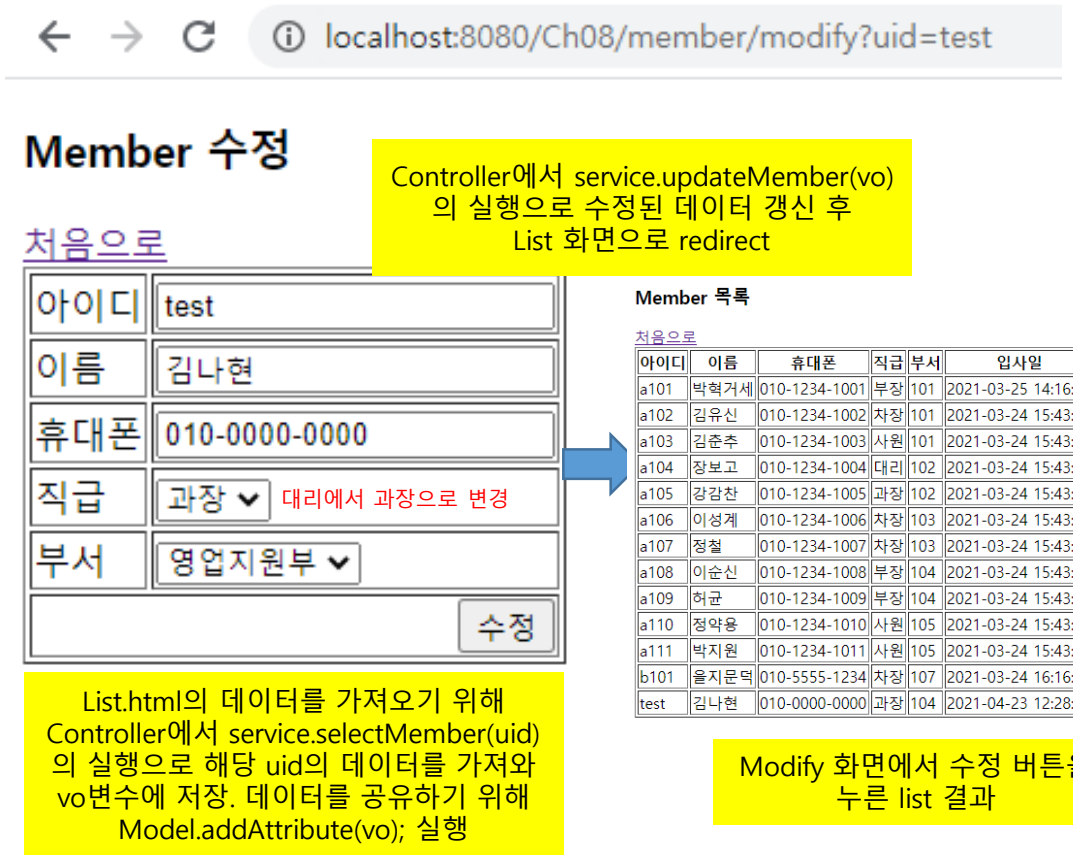
구현 화면

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>list</title>
</head>
<body>
    <h3>Member 목록</h3>
    <a th:href="@{/}">처음으로</a>
    <table border="1">
        <tr>
            <th>아이디</th>
            <th>이름</th>
            <th>휴대폰</th>
            <th>직급</th>
            <th>부서</th>
            <th>입사일</th>
            <th>관리</th>
        </tr>

        <tr th:each="member:${members}">
            <td th:text="${member.uid}">아이디</td>
            <td th:text="${member.name}">이름</td>
            <td th:text="${member.hp}">휴대폰</td>
            <td th:text="${member.pos}">직급</td>
            <td th:text="${member.dep}">부서</td>
            <td th:text="${member.rdate}">입사일</td>
            <td>
                <a th:href="@{/member/modify(uid=${member.uid})}">수정</a>
                <a th:href="@{/member/delete(uid=${member.uid})}">삭제</a>
            </td>
        </tr>
    </table>

</body>
</html>
```

Controller에서 service.selectMembers()의 실행으로 members에 대한 데이터 공유. 공유된 members를 member 변수로 각각 반복하여 list 출력.

코드

modify.html



구현 화면

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>modify</title>
</head>
<body>
    <h3>Member 수정</h3>
    <a th:href="@{/}">처음으로</a>
    <form th:action="@{/member/modify}" method="post">
        <table border="1">
            <tr>
                <td>아이디</td>
                <td><input type="text" name="uid" readonly th:value="${memberVo.uid}"/></td>
            </tr>
            <tr>
                <td>이름</td>
                <td><input type="text" name="name" th:value="${memberVo.name}"/></td>
            </tr>
            <tr>
                <td>휴대폰</td>
                <td><input type="text" name="hp" th:value="${memberVo.hp}"/></td>
            </tr>
            <tr>
                <td>직급</td>
                <td>
                    <select name="pos">
                        <option th:selected="${memberVo.pos eq '사원'}">사원</option>
                        <option th:selected="${memberVo.pos eq '대리'}">대리</option>
                        <option th:selected="${memberVo.pos eq '과장'}">과장</option>
                        <option th:selected="${memberVo.pos eq '차장'}">차장</option>
                        <option th:selected="${memberVo.pos eq '부장'}">부장</option>
                    </select>
                </td>
            </tr>
            <tr>
                <td>부서</td>
                <td>
                    <select name="dep">
                        <option value="101" th:selected="${memberVo.dep == 101}">영업1부</option>
                        <option value="102" th:selected="${memberVo.dep == 102}">영업2부</option>
                        <option value="103" th:selected="${memberVo.dep == 103}">영업3부</option>
                        <option value="104" th:selected="${memberVo.dep == 104}">영업지원부</option>
                        <option value="105" th:selected="${memberVo.dep == 105}">인사부</option>
                    </select>
                </td>
            </tr>
            <tr>
                <td colspan="2" align="right"><input type="submit" value="수정"/></td>
            </tr>
        </table>
    </form>
</body>
</html>
```

데이터를 받아오기 위해 th:selected 문법 사용

코드