

5. 통합 구현

이홍주

Spring Boot / Mybatis

```
<?xml version="1.0" encoding="UTF-8"?>
<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/xsd/maven-4.0.0.xsd">
<modelVersion>4.0.0</modelVersion>
<parent>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-parent</artifactId>
<version>2.4.5</version>
<relativePath/> <!-- lookup parent from repository -->
</parent>
<groupId>kr.co.ch08</groupId>
<artifactId>Ch08</artifactId>
<version>0.0.1-SNAPSHOT</version>
<packaging>war</packaging>
<name>Ch08</name>
<description>Demo project for Spring Boot</description>
<properties>
<java.version>1.8</java.version>
</properties>
<dependencies>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-jdbc</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-thymeleaf</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web</artifactId>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-starter-web-services</artifactId>
</dependency>
<dependency>
<groupId>org.mybatis.spring.boot</groupId>
<artifactId>mybatis-spring-boot-starter</artifactId>
<version>2.1.4</version>
</dependency>
<dependency>
<groupId>org.springframework.boot</groupId>
<artifactId>spring-boot-devtools</artifactId>
<scope>runtime</scope>
<optional>true</optional>
</dependency>
<dependency>
<groupId>mysql</groupId>
<artifactId>mysql-connector-java</artifactId>
<scope>runtime</scope>
</dependency>
<dependency>
<groupId>org.projectlombok</groupId>
<artifactId>lombok</artifactId>
<optional>true</optional>
</dependency>
</dependencies>
```

Pom.xml 에서 라이브러리 확인

```
1 #livereload Setting(Must install Chrome livereload plugin First)
2 spring.devtools.livereload.enabled=true
3 spring.thymeleaf.cache=false
4
5 #Context root Setting
6 server.servlet.context-path=/Ch08
7
8 #DB Setting
9 spring.datasource.url=jdbc:mysql://192.168.10.114:3306/lhj
0 spring.datasource.username=lhj
1 spring.datasource.password=1234
2 spring.datasource.driver-class-name=com.mysql.jdbc.Driver
3
4 #Mybatis Setting(Ch08Application add @MapperScan("kr.co.ch08.dao"))
5 mybatis.mapper-locations=classpath:mappers/**/*.xml
```

Application.properties에 thymeleaf와 db세팅 mybatis세팅 등을 해준다

```
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);
    }
```

```
}
```

Application클래스에 mapperScan을 하여 연결해준다

설정

```
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 String dep;
    private String rdate;
}
```

데이터베이스에 값을 받을
MemberVo 만들기

```
<?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},
            `hp`=#{hp},
            `pos`=#{pos},
            `dep`=#{dep}
        WHERE
            `uid`=#{uid};
    </update>
    <delete id="deleteMember">
        DELETE FROM `MEMBER` WHERE `uid`=#{uid};
    </delete>
</mapper>
```

Mapper안에 member.xml에 쿼리문
작성

Dao. Service . Controller

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

import java.util.List;

@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);
}
```

MemberDao에 member.xml에 있는 값을 가져오기

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

import java.util.List;

@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);
    }
}
```

메서드 MemberDao를 호출하기

```
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 "/member/register";
    }

    @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";
    }
}
```

MemberController에 최종 호출한다

Register 화면 구현

```
<!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>부장</option>
<option>과장</option>
<option>차장</option>
<option>대리</option>
<option>사원</option>
</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>
```

Member 등록

[처음으로](#)

아이디	<input type="text" value="ev088"/>
이름	<input type="text" value="뿌링클"/>
휴대폰	<input type="text" value="010-8253-8253"/>
직급	<input type="text" value="부장"/>
부서	<input type="text" value="영업지원부"/>
<input type="button" value="등록"/>	

아이디 이름 휴대폰 입력을 하고
직급과 부서는 선택을 하여 등록을
한다

Register 화면 구조 짜기
Thymeleaf으므로 th:를 붙이고 사용

List화면 구현

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
  <meta charset="UTF-8">
  <title>list</title>
</head>
<body>
  <h3>Member 목록</h3>
  <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>
```

List 화면 구조 짜기
Thymeleaf 이므로 th: 붙어준다

Member 목록

아이디	이름	휴대폰	직급	부서	입사일	관리
a101	박혁거세	010-1234-1001	부장	101	2021-03-24 15:02:26	수정 삭제
a1010	정약용	010-1234-1010	사원	105	2021-03-24 15:02:26	수정 삭제
a1011	박지원	010-1234-1011	사원	105	2021-03-24 15:02:26	수정 삭제
a102	김유신	010-1234-1002	차장	101	2021-03-24 15:02:26	수정 삭제
a103	김춘추	010-1234-1003	사원	101	2021-03-24 15:02:26	수정 삭제
a104	장보고	010-1234-1004	대리	102	2021-03-24 15:02:26	수정 삭제
a105	감감찬	010-1234-1005	과장	102	2021-03-24 15:02:26	수정 삭제
a106	이성계	010-1234-1006	차장	103	2021-03-24 15:02:26	수정 삭제
a107	정철	010-1234-1007	차장	103	2021-03-24 15:02:26	수정 삭제
a108	이순신	010-1234-1008	부장	104	2021-03-24 15:02:26	수정 삭제
a109	허균	010-1234-1009	부장	104	2021-03-24 15:02:26	수정 삭제
b101	을지문덕	010-5555-1234	사장	107	2021-03-24 15:43:56	수정 삭제
d101	가나다	010-1234-4321	부장	101	2021-04-19 17:20:17	수정 삭제
ev088	뿌링클	010-8253-8253	부장	104	2021-04-23 11:39:02	수정 삭제

리스트에 등록이 되었다.

Modify 화면 구성

```
<!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="${memberVo.hp}"/></td>
      </tr>
      <tr>
        <td>직급</td>
        <td>
          <select name="pos" th:value="${memberVo.pos}"/>
            <option>부장</option>
            <option>과장</option>
            <option>차장</option>
            <option>대리</option>
            <option>사원</option>
          </select>
        </td>
      </tr>
      <tr>
        <td>부서</td>
        <td>
          <select name="dep" th:value="${memberVo.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="수정"/></td>
      </tr>
    </table>
  </form>
</body>
</html>
```

Member 수정

처음으로

아이디	ev088
이름	뿌링클
휴대폰	010-9999-9999
직급	부장 ▼
부서	영업3부 ▼
<input type="button" value="수정"/>	

수정 버튼을 눌러 수정을 함

ev088	뿌링클	010-9999-9999	부장	103	2021-04-23 11:39:02	수정 삭제
-------	-----	---------------	----	-----	---------------------	---------------------------------------

휴대폰 번호와 부서 수정

Member 목록

아이디	이름	휴대폰	직급	부서	입사일	관리
a101	박혁거세	010-1234-1001	부장	101	2021-03-24 15:02:26	수정 삭제
a1010	정약용	010-1234-1010	사원	105	2021-03-24 15:02:26	수정 삭제
a1011	박지원	010-1234-1011	사원	105	2021-03-24 15:02:26	수정 삭제
a102	김유신	010-1234-1002	차장	101	2021-03-24 15:02:26	수정 삭제
a103	김춘추	010-1234-1003	사원	101	2021-03-24 15:02:26	수정 삭제
a104	장보고	010-1234-1004	대리	102	2021-03-24 15:02:26	수정 삭제
a105	감감찬	010-1234-1005	과장	102	2021-03-24 15:02:26	수정 삭제
a106	이성계	010-1234-1006	차장	103	2021-03-24 15:02:26	수정 삭제
a107	정철	010-1234-1007	차장	103	2021-03-24 15:02:26	수정 삭제
a108	이순신	010-1234-1008	부장	104	2021-03-24 15:02:26	수정 삭제
a109	허균	010-1234-1009	부장	104	2021-03-24 15:02:26	수정 삭제
b101	을지문덕	010-5555-1234	사장	107	2021-03-24 15:43:56	수정 삭제
d101	가나다	010-1234-4321	부장	101	2021-04-19 17:20:17	수정 삭제

삭제 버튼을 눌러
Member 목록에서 ev088
뿌링클 삭제가 됨

Modify 화면 구조 짜기