

데이터 입출력 구현

김 나 현

프로젝트 구조

```
▼ IDE > Ch05 [Spring master]
  ▼ src/main/java
    ▼ kr.co.ch05.controller
      > IndexController.java
      > MemberController.java
      > UserController.java
    ▼ kr.co.ch05.dao
      > MemberDao.java
      > UserDao.java
    ▼ kr.co.ch05.service
      > MemberService.java
      > UserService.java
    ▼ kr.co.ch05.vo
      > MemberVo.java
      > UserVo.java
  ▼ src/main/resources
    ▼ mapper
      member.xml
      user.xml
    META-INF
    log4j.xml
    mybatis-context.xml
    spring-context.xml
```

```
> src/test/java
> src/test/resources
> JRE System Library [JavaSE-1.8]
> Maven Dependencies
▼ src
  ▼ main
    ▼ webapp
      resources
      ▼ WEB-INF
        ▼ views
          ▼ member
            list.jsp
            > modify.jsp
            > register.jsp
          > user
            index.jsp
            web.xml
      test
```

web.xml

```
<!-- Spring Front-Controller(Dispatcher) 등록 -->
<servlet>
    <servlet-name>DispatcherServlet</servlet-name>
    <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
    <init-param> <!-- 생성자 -->
        <param-name>contextConfigLocation</param-name>
        <param-value>classpath:spring-context.xml</param-value>
    </init-param>
</servlet>

<servlet-mapping>
    <servlet-name>DispatcherServlet</servlet-name>
    <url-pattern>/</url-pattern>
</servlet-mapping>
```

mybatis-context

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE configuration
    PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
    <mappers>
        <mapper resource="./mapper/user.xml"/>
        <mapper resource="./mapper/member.xml"/>
    </mappers>
</configuration>
```

Client가 페이지를 요청하면 web.xml에서 등록한
FontController가 먼저 실행됨. mybatis-context에서
mapper 파일 설정.

spring.xml

```
<!-- 어노테이션 기반 컴포넌트 스캔 -->
<context:component-scan base-package="kr.co.ch05"></context:component-scan>

<!-- 뷰 처리를 위한 ViewResolver 설정 -->
<bean id="viewResolver" class="org.springframework.web.servlet.view.InternalResourceViewResolver">
    <property name="prefix" value="/WEB-INF/views/"></property> <!-- 접두어 -->
    <property name="suffix" value=".jsp"></property> <!-- 접미어 -->
    <!-- viewResolver가 접두어, 접미어들을 붙여줄까다 -->
</bean>

<!-- Spring MVC(Web)에서 사용되는 Annotation 활성화를 위해 설정 -->
<mvc:annotation-driven></mvc:annotation-driven>
```

```
<!-- 데이터베이스 설정 -->
<bean id="basicDataSource" class="org.apache.commons.dbcp2.BasicDataSource" destroy-method="close">
    <property name="driverClassName" value="com.mysql.jdbc.Driver"></property>
    <property name="url" value="jdbc:mysql://192.168.10.114:3306/knh"></property>
    <property name="username" value="knh"></property>
    <property name="password" value="1234"></property>
</bean>

<!-- JDBC 설정 -->
<!-- 29라인 데이터베이스 설정 id 참고 -->
<bean id="jdbcTemplate" class="org.springframework.jdbc.core.JdbcTemplate">
    <property name="dataSource" ref="basicDataSource"></property>
</bean>

<!-- Mybatis 설정 -->
<bean id="sqlSessionFactoryBean" class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="dataSource" ref="basicDataSource"></property>
    <property name="configLocation" value="classpath:mybatis-context.xml"></property>
</bean>

<!-- Mybatis-Spring 설정 -->
<!-- 생성자? -->
<bean id="sqlSessionTemplate" class="org.mybatis.spring.SqlSessionTemplate">
    <constructor-arg ref="sqlSessionFactoryBean"></constructor-arg>
</bean>
```

DB 연결을 위해 mybatis ORM 사용 설정

MemberVo.java

```
package kr.co.ch05.vo;

public class MemberVo {

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

    public String getUid() {
        return uid;
    }
    public void setUid(String uid) {
        this.uid = uid;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public String getHp() {
        return hp;
    }
    public void setHp(String hp) {
        this.hp = hp;
    }
    public String getPos() {
        return pos;
    }
    public void setPos(String pos) {
        this.pos = pos;
    }
    public int getDep() {
        return dep;
    }
    public void setDep(int dep) {
        this.dep = dep;
    }
    public String getRdate() {
        return rdate;
    }
    public void setRdate(String rdate) {
        this.rdate = rdate;
    }
}
```

DB의 컬럼값을 받아줄
MemberVo

member.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="mapper.member">

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

    <select id="SELECT_MEMBERS" resultType="kr.co.ch05.vo.MemberVo"> <!-- 총 ??개(Member개수)의 vo를 만들어서 List! -->
        SELECT * FROM `MEMBER`;
    </select>

    <select id="SELECT_MEMBER" resultType="kr.co.ch05.vo.MemberVo">
        SELECT * FROM `MEMBER` WHERE `uid`=#{uid};
    </select>

    <update id="UPDATE_MEMBER">
        UPDATE `MEMBER` SET
            `name`=#{name},
            `hp`=#{hp},
            `pos`=#{pos},
            `dep`=#{dep}
        WHERE
            `uid`=#{uid};
    </update>

    <delete id="DELETE_MEMBER">
        DELETE FROM `MEMBER` WHERE `uid`=#{uid};
    </delete>

</mapper>
```

Mybatis-context.xml에서 지정해준 member.xml에
시스템에서 사용할 쿼리문 저장.

MemberDao.java

```
package kr.co.ch05.dao;

import java.util.List;

@Repository
public class MemberDao {

    //Mybatis 객체
    @Inject
    private SqlSessionTemplate mybatis;

    public void insertMember(MemberVo vo) {
        mybatis.insert("mapper.member.INSERT_MEMBER", vo); /* namespace:n
    }

    public MemberVo selectMember(String uid) {
        //외 MEMBERVo
        return mybatis.selectOne("mapper.member.SELECT_MEMBER", uid);
    }

    public List<MemberVo> selectMembers() {
        return mybatis.selectList("mapper.member.SELECT_MEMBERS");
    }

    public void updateMember(MemberVo vo) {
        mybatis.update("mapper.member.UPDATE_MEMBER", vo);
    }

    public void deleteMember(MemberVo vo) {
        mybatis.delete("mapper.member.DELETE_MEMBER", vo);
    }
}
```

member.xml에 저장되어 있는 쿼리문을 사용하여
mybatis 객체에 데이터 불러오기

MemberService.java

```
package kr.co.ch05.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(MemberVo vo) {
        dao.deleteMember(vo);
    }
}
```

MemberDao의 메소드를 호출.
Controller에서 최종적으로 사용.

MemberController.java

```
package kr.co.ch05.controller;

import java.util.List;

@Controller
public class MemberController {

    @Autowired
    private MemberService service;

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

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

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

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

        // View에서 데이터 출력을 위해 model객체로 데이터 공유(JSP에서는 req로 받던거임)
        model.addAttribute("members", members);

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

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

    MemberVo vo = service.selectMember(uid);

    // 공유객체명을 지정 안할 경우 해당 객체의 소문자로 시작하는 타입이름으로 공유
    model.addAttribute(vo);

    return "/member/modify";
}

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

    service.updateMember(vo);

    return "redirect:/member/list";
}

@GetMapping("/member/delete")
public String delete(MemberVo vo) {
    service.deleteMember(vo);
    return "redirect:/member/list";
}
}
```

쿼리문(메소드) 사용을 위해 MemberService의 service 객체 생성.
Client가 요청한 페이지에 대한 작업을 진행.

Index.jsp

Spring Mybatis 실습하기

[User 등록](#)

[User 목록](#)

[Member 등록](#)

[Member 목록](#)

구현 화면

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>index</title>
</head>
<body>
<h3>Spring Mybatis 실습하기</h3>

<p>
<a href="/ch05/user/register">User 등록</a><br/>
<a href="/ch05/user/list">User 목록</a><br/>
</p>

<p>
<a href="/ch05/member/register">Member 등록</a><br/>
<a href="/ch05/member/list">Member 목록</a><br/>
</p>

</body>
</html>
```

코드

register.jsp

← → ↻ ⓘ localhost:8080/ch05/member/register

Member 등록

아이디	<input type="text" value="test"/>
이름	<input type="text" value="김나현"/>
휴대폰	<input type="text" value="010-0000-0000"/>
직급	<input type="text" value="대리"/>
부서	<input type="text" value="영업지원부"/>
<input type="button" value="등록"/>	

등록할 데이터 입력

구현 화면

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>register</title>
</head>
<body>
  <h3>Member 등록</h3>
  <form action="/ch05/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</option>
            <option>대리</option>
            <option>과장</option>
            <option>자장</option>
            <option>부장</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" style="text-align: right;><input type="submit" value="등록"/></td>
      </tr>
    </table>
  </form>
</body>
</html>
```

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

코드

list.jsp

localhost:8080/ch05/member/list

Member 목록

Register.jsp에서 member를 등록한 결과 list

아이디	이름	휴대폰	직급	부서	입사일	관리
a101	박혁거세	010-1234-1001	부장	101	2021-03-25 14:16:57.0	수정 삭제
a102	김유신	010-1234-1002	차장	101	2021-03-24 15:43:46.0	수정 삭제
a103	김춘추	010-1234-1003	사원	101	2021-03-24 15:43:46.0	수정 삭제
a104	장보고	010-1234-1004	대리	102	2021-03-24 15:43:46.0	수정 삭제
a105	강감찬	010-1234-1005	과장	102	2021-03-24 15:43:46.0	수정 삭제
a106	이성계	010-1234-1006	차장	103	2021-03-24 15:43:46.0	수정 삭제
a107	정철	010-1234-1007	차장	103	2021-03-24 15:43:46.0	수정 삭제
a108	이순신	010-1234-1008	부장	104	2021-03-24 15:43:46.0	수정 삭제
a109	허균	010-1234-1009	부장	104	2021-03-24 15:43:46.0	수정 삭제
a110	정약용	010-1234-1010	사원	105	2021-03-24 15:43:46.0	수정 삭제
a111	박지원	010-1234-1011	사원	105	2021-03-24 15:43:46.0	수정 삭제
b101	을지문덕	010-5555-1234	사장	107	2021-03-24 16:16:14.0	수정 삭제
test	김나현	010-0000-0000	대리	104	2021-04-16 10:10:49.0	수정 삭제

대리에서 과장으로 승진 예정

구현 화면

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

```

<%@ 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>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>
        <c:forEach var="member" items="${members}">
            <tr>
                <td>${member.uid}</td>
                <td>${member.name}</td>
                <td>${member.hp}</td>
                <td>${member.pos}</td>
                <td>${member.dep}</td>
                <td>${member.rdate}</td>
                <td>
                    <a href="/ch05/member/modify?uid=${member.uid}">수정</a>
                    <a href="/ch05/member/delete?uid=${member.uid}">삭제</a>
                </td>
            </tr>
        </c:forEach>
    </table>
</body>
</html>

```

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

코드

modify.jsp

← → ↻ ⓘ localhost:8080/ch05/member/modify?uid=test

Member 수정

Controller에서 service.updateMember(vo)
의 실행으로 수정된 데이터 갱신 후
List 화면으로 redirect

처음으로

아이디	test
이름	김나현
휴대폰	010-0000-0000
직급	과장 ▼ 대리에서 과장으로 변경
부서	영업지원부 ▼
수정	



Member 목록

아이디	이름	휴대폰	직급	부서	입사일	관리
a101	박혁거세	010-1234-1001	부장	101	2021-03-25 14:16:57.0	수정 삭제
a102	김유신	010-1234-1002	차장	101	2021-03-24 15:43:46.0	수정 삭제
a103	김춘추	010-1234-1003	사원	101	2021-03-24 15:43:46.0	수정 삭제
a104	장보고	010-1234-1004	대리	102	2021-03-24 15:43:46.0	수정 삭제
a105	강감찬	010-1234-1005	과장	102	2021-03-24 15:43:46.0	수정 삭제
a106	이성계	010-1234-1006	차장	103	2021-03-24 15:43:46.0	수정 삭제
a107	정철	010-1234-1007	차장	103	2021-03-24 15:43:46.0	수정 삭제
a108	이순신	010-1234-1008	부장	104	2021-03-24 15:43:46.0	수정 삭제
a109	허균	010-1234-1009	부장	104	2021-03-24 15:43:46.0	수정 삭제
a110	정약용	010-1234-1010	사원	105	2021-03-24 15:43:46.0	수정 삭제
a111	박지원	010-1234-1011	사원	105	2021-03-24 15:43:46.0	수정 삭제
b101	울지문덕	010-5555-1234	사장	107	2021-03-24 16:16:14.0	수정 삭제
test	김나현	010-0000-0000	과장	104	2021-04-16 10:10:49.0	수정 삭제

List.jsp의 데이터를 가져오기 위해
Controller에서 service.selectMember(uid)
의 실행으로 해당 uid의 데이터를 가져와
vo 변수에 저장. 데이터를 공유하기 위해
Model.addAttribute(vo); 실행

Modify 화면에서 수정 버튼을
누른 list 결과

구현 화면

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>modify</title>
</head>
<body>
<h3>Member 수정</h3>
<a href="/ch05">처음으로</a>
<form action="/ch05/member/modify" method="post">
<table border="1">
<tr>
<td>아이디</td>
<td><input type="text" name="uid" readonly value="${memberVo.uid}"/></td>
</tr>
<tr>
<td>이름</td>
<td><input type="text" name="name" value="${memberVo.name}"/></td>
</tr>
<tr>
<td>휴대폰</td>
<td><input type="text" name="hp" value="${memberVo.hp}"/></td>
</tr>
<tr>
<td>직급</td>
<td>
<select name="pos">
<option ${memberVo.pos eq '사원' ? 'selected' : 'null'}>사원</option>
<option ${memberVo.pos eq '대리' ? 'selected' : 'null'}>대리</option>
<option ${memberVo.pos eq '과장' ? 'selected' : 'null'}>과장</option>
<option ${memberVo.pos eq '차장' ? 'selected' : 'null'}>차장</option>
<option ${memberVo.pos eq '부장' ? 'selected' : 'null'}>부장</option>
</select>
</td>
</tr>
<tr>
<td>부서</td>
<td>
<select name="dep">
<option value="101" ${memberVo.dep eq 101 ? 'selected' : 'null'}>영업1부</option>
<option value="102" ${memberVo.dep eq 102 ? 'selected' : 'null'}>영업2부</option>
<option value="103" ${memberVo.dep eq 103 ? 'selected' : 'null'}>영업3부</option>
<option value="104" ${memberVo.dep eq 104 ? 'selected' : 'null'}>영업지원부</option>
<option value="105" ${memberVo.dep eq 105 ? 'selected' : 'null'}>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 삼항연산자 사용

코드