

# 4. 데이터 입출력 구현

---

이홍주

# 설정

## Mybatis를 사용하기 위해서 설정

```
<!-- 추가 필수 라이브러리 -->
<dependency>
  <groupId>org.aspectj</groupId>
  <artifactId>aspectjweaver</artifactId>
  <version>1.9.6</version>
</dependency>

<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>5.1.49</version>
</dependency>

<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis</artifactId>
  <version>3.5.6</version>
</dependency>

<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis-spring</artifactId>
  <version>2.0.6</version>
</dependency>

<dependency>
  <groupId>org.apache.commons</groupId>
  <artifactId>commons-dbcp2</artifactId>
  <version>2.8.0</version>
</dependency>

<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-jdbc</artifactId>
  <version>${org.springframework-version}</version>
</dependency>
```

Pom.xml에서 추가 라이브러리를 입력하기

### Spring-context.xml

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

### Mybatis-context.xml

```
<?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>
```

# 설정2

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

public class MemberVo {

    private String uid;
    private String name;
    private String hp;
    private String pos;
    private String 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 String getDep() {
        return dep;
    }
    public void setDep(String dep) {
        this.dep = dep;
    }
    public String getRdate() {
        return rdate;
    }
    public void setRdate(String rdate) {
        this.rdate = 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="mapper.member">

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

    <select id="SELECT_MEMBERS" resultType="kr.co.ch05.vo.MemberVo">
        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},
            `hp`=#{hp},
            `pos`=#{pos},
            `dep`=#{dep}
        WHERE
            `uid`=#{uid};
    </update>

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

Mybatis-context.xml 지정해준 쿼리문 만들기

# 설정3

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

import java.util.List;

@Repository
public class MemberDao {

    @Inject
    private SqlSessionTemplate mybatis;

    public void insertMember(MemberVo vo) {
        mybatis.insert("mapper.member.INSERT_MEMBER", vo);
    }
    public MemberVo selectMember(String uid) {
        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);
    }
}
```

MemberDao에 member.xml에 있는 값을 가져오기

```
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);
    }
}
```

메서드 MemberDao를 호출하기

```
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();
        model.addAttribute("members", members);
        return "/member/list";
    }

    @GetMapping("/member/modify")
    public String modify(String uid, Model model) {
        MemberVo vo = service.selectMember(uid);
        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";
    }
}
```

MemberController에 최종 호출한다

# Register 화면 구현

```
<%@ 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>
<a href="/ch05">처음으로</a>
<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>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>
```

Register 화면 구조 짜기

## Member 등록

처음으로

아이디	<input type="text" value="c101"/>
이름	<input type="text" value="갈몽이"/>
휴대폰	<input type="text" value="010-0805-0805"/>
직급	<input type="text" value="부장"/>
부서	<input type="text" value="영업1부"/>
<input type="button" value="등록"/>	

아이디 이름 휴대폰 입력을 하고  
직급과 부서는 선택을 하여 등록을 한다

# List화면 구현

```
<%@ 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>
```

List 화면 구조 짜기 Member변수를 각각 주기

## Member 목록

아이디	이름	휴대폰	직급	부서	입사일	관리
a101	박혁거세	010-1234-1001	부장	101	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a1010	정약용	010-1234-1010	사원	105	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a1011	박지원	010-1234-1011	사원	105	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a102	김유신	010-1234-1002	차장	101	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a103	김춘추	010-1234-1003	사원	101	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a104	장보고	010-1234-1004	대리	102	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a105	감감찬	010-1234-1005	과장	102	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a106	이성계	010-1234-1006	차장	103	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a107	정철	010-1234-1007	차장	103	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a108	이순신	010-1234-1008	부장	104	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a109	허균	010-1234-1009	부장	104	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
b101	을지문덕	010-5555-1234	사장	107	2021-03-24 15:43:56.0	<a href="#">수정</a> <a href="#">삭제</a>
c101	갈몽이	010-0805-0805	부장	101		<a href="#">수정</a> <a href="#">삭제</a>

# Modify 화면 구성

```

<%@ 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="${memberVo.hp}"/></td>
</tr>
<tr>
<td>직급</td>
<td>
<select name="pos" value="${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" value="${memberVo.dep}">
<option value="101">영역1부</option>
<option value="102">영역2부</option>
<option value="103">영역3부</option>
<option value="104">영역지원부</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	c101
이름	갈몽이
휴대폰	010-0805-0505
직급	부장 ▼
부서	영역지원부 ▼
수정	

수정 버튼을 눌러 수정을 함

c101 갈몽이 010-0805-0505 부장 104

휴대폰 번호와 부서 수정

## Member 목록

아이디	이름	휴대폰	직급	부서	입사일	관리
a101	박혁거세	010-1234-1001	부장	101	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a1010	정약용	010-1234-1010	사원	105	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a1011	박지원	010-1234-1011	사원	105	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a102	김유신	010-1234-1002	차장	101	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a103	김준추	010-1234-1003	사원	101	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a104	장보고	010-1234-1004	대리	102	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a105	감감찬	010-1234-1005	과장	102	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a106	이성계	010-1234-1006	차장	103	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a107	정철	010-1234-1007	차장	103	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a108	이순신	010-1234-1008	부장	104	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
a109	허균	010-1234-1009	부장	104	2021-03-24 15:02:26.0	<a href="#">수정</a> <a href="#">삭제</a>
b101	율지문덕	010-5555-1234	사장	107	2021-03-24 15:43:56.0	<a href="#">수정</a> <a href="#">삭제</a>

삭제 버튼을 눌러  
Member 목록에서  
c101갈몽이 삭제가 됨

Modify 화면 구조 짜기 Member변수를 각각 주기