

# DEVELOP CRUD PROGRAM FOR MEMBER TABLE USING SPRING AND MYBATIS

Stella Kim

April 16<sup>th</sup>, 2021

# INDEX

---

Project Structure

---

MemberVo

---

Mapper

---

MemberDao

---

MembersService

---

MemberController

---

INSERT Operation

---

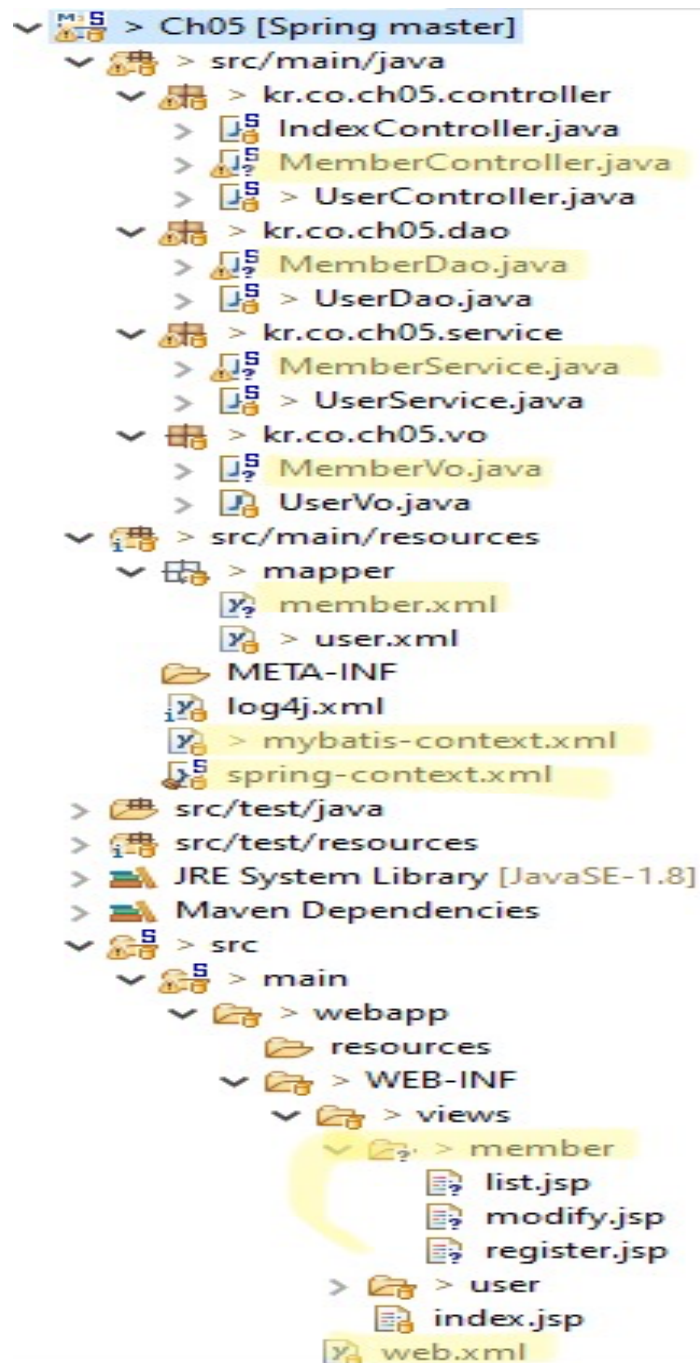
SELECT Operation

---

UPDATE Operation

---

DELETE Operation



## PROJECT STRUCTURE

- Create a Spring MVC project and configure necessary MyBatis dependencies to pom.xml file.
- Create MyBatis configuration file as “mybatis-context.xml” in src/main/resources folder.
- Create the Vo, Controller, Dao, and Service classes in each package.
- Create the view components including list, modify, and register pages in member folder.

# VO

```
CREATE TABLE `MEMBER` (  
  `uid` VARCHAR(10) PRIMARY KEY,  
  `name` VARCHAR(10),  
  `hp` CHAR(13),  
  `pos` VARCHAR(10),  
  `dep` TINYINT,  
  `rdate` DATETIME  
);
```

Create the table `MEMBER` in the SQL database including 6 fields: user id as a primary key, name, phone number, position, department, and start date.

```
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;  
    }  
}
```

Make a Vo class containing all of the variables along with setter and getter methods.

# MAPPER

```
<?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>
```

Set a mapper in mybatis-context.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">
    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>
```

Write all the SQL Query Statements  
in member.xml under mapper package.

# DAO

@Repository  
To access the required  
data from the database

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

import java.util.List;

import javax.inject.Inject;

import org.mybatis.spring.SqlSessionTemplate;
import org.springframework.stereotype.Repository;

import kr.co.ch05.vo.MemberVo;

@Repository
public class MemberDao {
    //MyBatis Object
    @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);
    }
}
```

# SERVICE

@Service

To carry out different  
transactions for each  
function

```
package kr.co.ch05.service;

import java.util.List;

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

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

```

package kr.co.ch05.controller;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.PostMapping;

import kr.co.ch05.service.MemberService;
import kr.co.ch05.vo.MemberVo;

@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";
    }
}

```

# CONTROLLER

@Controller

To navigate between the different views



# INSERT

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Register</title>
</head>
<body>
<h3>Register an employee</h3>
<a href="/ch05">Back to index</a>
<form action="/ch05/member/register" method="post">
<table border="1">
<tr>
<td>ID</td>
<td><input type="text" name="uid"/></td>
</tr>
<tr>
<td>Name</td>
<td><input type="text" name="name"/></td>
</tr>
<tr>
<td>Phone</td>
<td><input type="text" name="hp"/></td>
</tr>
<tr>
<td>Position</td>
<td>
<select name="pos">
<option>associate</option>
<option>manager class2</option>
<option>manager class1</option>
<option>deputy manager</option>
<option>head manager</option>
</select>
</td>
</tr>
</table>
</form>
</body>
</html>
```

```
<tr>
<td>Department</td>
<td>
<select name="dep">
<option value="101">1</option>
<option value="102">2</option>
<option value="103">3</option>
<option value="104">4</option>
<option value="105">5</option>
</select>
</td>
</tr>
<tr>
<td colspan="2" align="right"><input type="submit" value="Register"/></td>
</tr>
</table>
</form>
</body>
</html>
```

## Spring MyBatis Index

[Register an user.](#) [View the list of users.](#)

[Add a new employee.](#) [View the list of employees.](#)

Go to register page by clicking "Add a new employee" from index page.

## Register an employee

[Back to index](#)

ID	<input type="text"/>
Name	<input type="text"/>
Phone	<input type="text"/>
Position	associate ▼
Department	1 ▼
<input type="button" value="Register"/>	

SELECT

```
<%@ 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>Employees List</h3>
<table border="1">
<tr>
<th>ID</th>
<th>Name</th>
<th>Phone</th>
<th>Position</th>
<th>Department</th>
<th>Start Date</th>
<th>Manage</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}">Edit</a>
<a href="/ch05/member/delete?uid=${member.uid}">Delete</a>

```

## Spring MyBatis Index

[Register an user.](#) [View the list of users.](#)

[Add a new employee.](#) [View the list of employees.](#)

Go to list page by clicking  
“View the list of employees”  
from index page.

## Employees List

ID	Name	Phone	Position	Department	Start Date	Manage
a102	김유신	010-1234-1002	차장	101	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
a103	김준추	010-1234-1003	사원	101	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
a104	장보고	010-1234-1004	대리	102	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
a105	강감찬	010-1234-1005	과장	102	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
a106	이성계	010-1234-1006	차장	103	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
a107	정철	010-1234-1007	차장	103	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
a108	이순신	010-1234-1008	부장	104	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
a109	허균	010-1234-1009	부장	104	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
a110	정약용	010-1234-1010	사원	105	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
a111	박지원	010-1234-1011	사원	105	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
b101	윤지문덕	010-5555-1234	사장	107	2021-03-24 15:37:28.0	<a href="#">Edit</a> <a href="#">Delete</a>
b102	김김김	010-9999-9999	associate	101	2021-04-16 15:01:50.0	<a href="#">Edit</a> <a href="#">Delete</a>

UPDATE/DELETE

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Modify</title>
</head>
<body>
<h3>Modify the info</h3>
<a href="/ch05">Back to index</a>
<form action="/ch05/member/modify" method="post">
<table border="1">
<tr>
<td>ID</td>
<td><input type="text" name="uid" readonly value="${memberVo.uid}"/></td>
</tr>
<tr>
<td>Name</td>
<td><input type="text" name="name" value="${memberVo.name}"/></td>
</tr>
<tr>
<td>Phone</td>
<td><input type="text" name="hp" value="${memberVo.hp}"/></td>
</tr>
<tr>
<td>Position</td>
<td>
<select name="pos">
<option>associate</option>
<option>manager class2</option>
<option>manager class1</option>
<option>deputy manager</option>
<option>head manager</option>
</select>
</td>
</tr>
</table>
</form>
</body>
</html>
```

## Employees List

ID	Name	Phone	Position	Department	Start Date	Manage
a102	김유신	010-1234-1002	차장	101	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
a103	김준추	010-1234-1003	사원	101	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
a104	장보고	010-1234-1004	대리	102	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
a105	강감찬	010-1234-1005	과장	102	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
a106	이성계	010-1234-1006	차장	103	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
a107	정철	010-1234-1007	차장	103	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
a108	이순신	010-1234-1008	부장	104	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
a109	허균	010-1234-1009	부장	104	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
a110	정약용	010-1234-1010	사원	105	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
a111	박지원	010-1234-1011	사원	105	2021-03-24 14:44:02.0	<a href="#">Edit</a> <a href="#">Delete</a>
b101	을지문덕	010-5555-1234	사장	107	2021-03-24 15:37:28.0	<a href="#">Edit</a> <a href="#">Delete</a>
b102	김김김	010-9999-9999	associate	101	2021-04-16 15:01:50.0	<a href="#">Edit</a> <a href="#">Delete</a>

Redirect to list page after performing the delete operation.

Go to modify page by clicking “Edit” of the specific employee from the list page.

### Modify the info

[Back to index](#)

ID	a102
Name	김유신
Phone	010-1234-1002
Position	associate ▼
Department	1 ▼
<div>Save changes</div>	