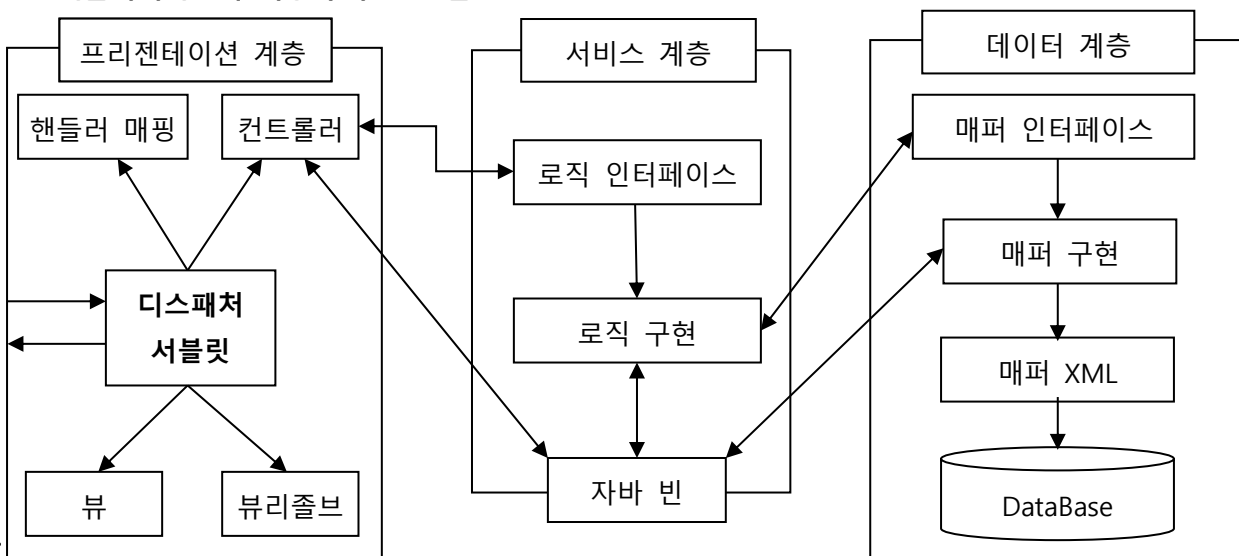## 11. Spring-Mybatis 연동 예제

(1) 테이블 생성

```
-- 부서테이블 생성
create table department(
        dept_id varchar2(10) not null,
        dept_name varchar2(25) not null,
        dept_tel varchar2(15)
);
comment on table   department is '부서테이블 정보';
comment on column department.dept_id is '부서번호';
comment on column department.dept_name is '부서명';
comment on column department.dept_tel is '부서별 전화번호';
```
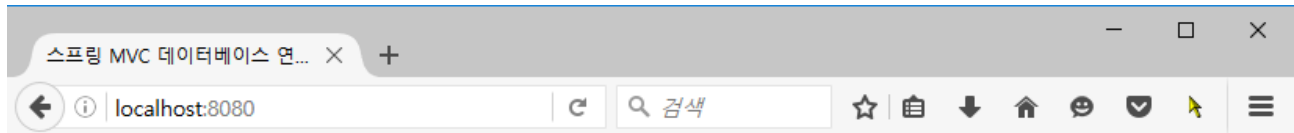
2) 개발 순서

- 테이블을 생성하고, 데이터를 저장한다.
- 프로젝트에 필요한 폴더를 생성한다.
- 모델에 관한 공통의 자바빈을 생성한다.
- 데이터 계층의 SQL 매퍼 XML 파일을 생성한다.
- 데이터 계층의 매퍼 인터페이스와 매퍼 인터페이스를 구현한다.
- 서비스 계층의 로직 인터페이스와 로직 인터페이스를 구현한다.
- 컨트롤러를 작성한다.
- 뷰를 작성한다.
- 스프링 환경 설정 파일에 dataSource, 매퍼 등을 설정한다.
- 웹 어플리케이션 환경 설정 파일(web.xml)에 디스패처 서블릿을 설정한다.
- 실행에 필요한 웹 페이지를 작성하고 실행한다.
- 실행 결과를 컴토하고, 오류에 대하여 디버깅한다.

- **웹 애플리케이션의 계층과 구현 모델**



185

코딩 후 결과화면



[학과 데이터]

① "SpringMyBatisExam" 프로젝트 생성(New -> Spring Legacy Project)
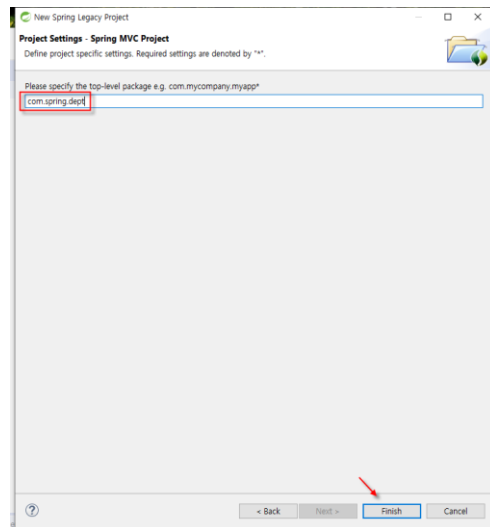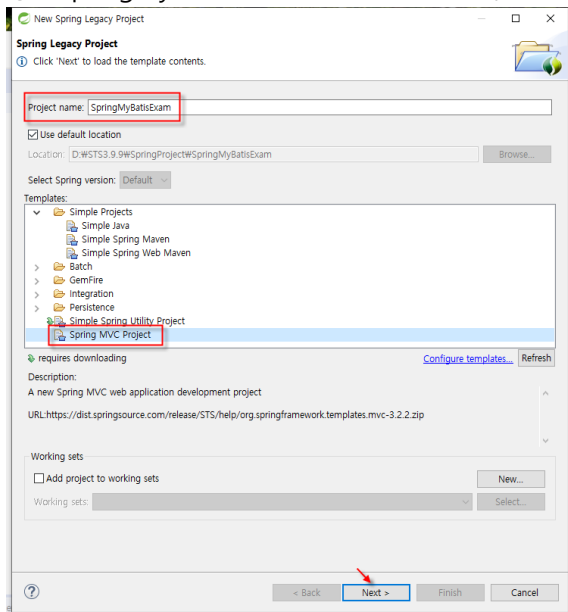


com.spring.dept
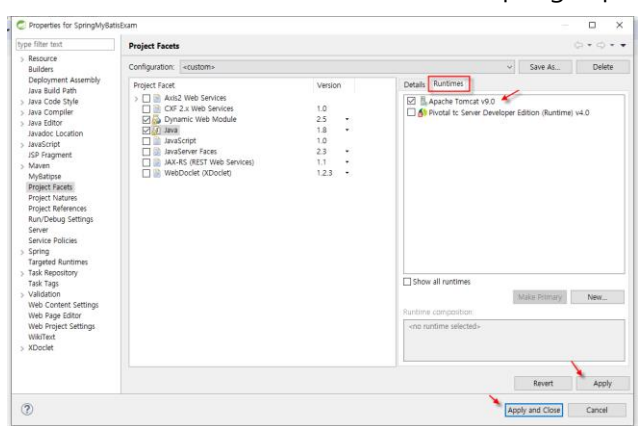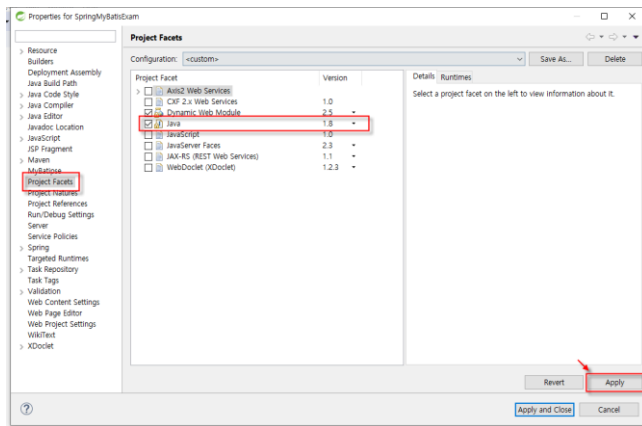


오라클 드라이버 연동
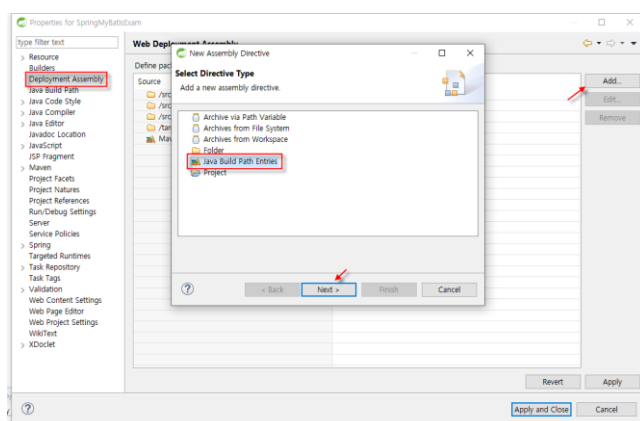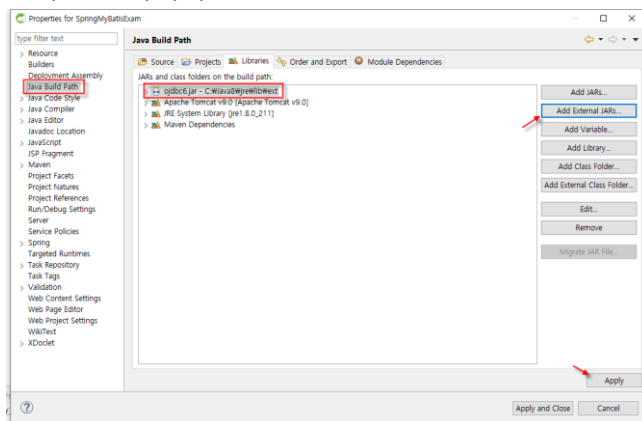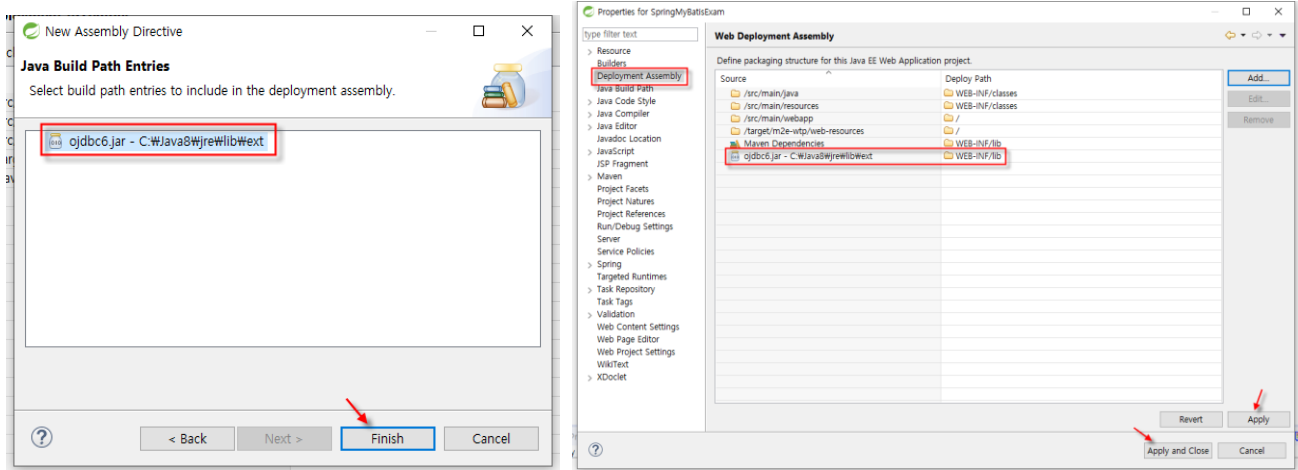
② 스프링과 MyBatis 연동에 필요한 라이브러리을 내려 받기 위해 pom.xml 수정

pom.xml

```
생략~
                <properties>
                <java-version>1.8</java-version>
                <org.springframework-version>5.0.7.RELEASE</org.springframework-version>
                <org.aspectj-version>1.6.10</org.aspectj-version>
                <org.slf4j-version>1.6.6</org.slf4j-version>
        </properties>
        <dependencies>
                <!-- MyBatis -->
                <!-- https://mvnrepository.com/artifact/org.mybatis/mybatis -->
                <dependency>
                        <groupId>org.mybatis</groupId>
                        <artifactId>mybatis</artifactId>
                        <version>3.5.2</version>
                </dependency>

                <!-- MyBatis Spring -->
                <dependency>
                        <groupId>org.mybatis</groupId>
                        <artifactId>mybatis-spring</artifactId>
                        <version>2.0.2</version>
                </dependency>

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

```
                </dependency>

                <!-- DBCP -->
                <dependency>
                        <groupId>commons-dbcp</groupId>
                        <artifactId>commons-dbcp</artifactId>
                        <version>1.4</version>
                </dependency>

                <!-- Spring -->
                <dependency>
생략
```

③ 웹 애플리케이션 환경 설정 파일에서 설정 수정

| src/main/webapp/WEB-INF/web.xml 웹 애플리케이션 환경 설정 파일에 추가 |
| --- |

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://java.sun.com/xml/ns/javaee http://java.sun.com/xml/ns/javaee/web-
app_2_5.xsd">

        <welcome-file-list>
                <welcome-file>index.jsp</welcome-file>
        </welcome-file-list>

        <!-- The definition of the Root Spring Container shared by all Servlets and Filters -->
        <context-param>
                <param-name>contextConfigLocation</param-name>
                <param-value>/WEB-INF/spring/root-context.xml</param-value>
        </context-param>

        <!-- Creates the Spring Container shared by all Servlets and Filters -->
        <listener>
                <listener-class>org.springframework.web.context.ContextLoaderListener</listener-
class>
        </listener>

        <!-- Processes application requests -->
        <servlet>
                <servlet-name>appServlet</servlet-name>
                <servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>
```

```xml
                <init-param>
                        <param-name>contextConfigLocation</param-name>
                        <param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>
                </init-param>
                <load-on-startup>1</load-on-startup>
        </servlet>

        <servlet-mapping>
                <servlet-name>appServlet</servlet-name>
                <url-pattern>*.do</url-pattern>
        </servlet-mapping>

        <!-- =============== 한글 처리 설정================= -->
        <filter>
                <filter-name>encodingFilter</filter-name>
                <filter-class>
                                org.springframework.web.filter.CharacterEncodingFilter
                </filter-class>
                <init-param>
                        <param-name>encoding</param-name>
                        <param-value>utf-8</param-value>
                </init-param>
                <init-param>
                        <param-name>forceEncoding</param-name>
                        <param-value>true</param-value>
                </init-param>
        </filter>
        <filter-mapping>
                <filter-name>encodingFilter</filter-name>
                <url-pattern>/*</url-pattern>
        </filter-mapping>
</web-app>
```

④ 스프링 설정 파일 생성 - 1

| webapp/WEB-INF/spring/appServlet/**servlet-context.xml** |
| --- |
| `<?xml version="1.0" encoding="UTF-8"?>`<br>`<beans:beans xmlns="http://www.springframework.org/schema/mvc"`<br>    `xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"`<br>    `xmlns:beans="http://www.springframework.org/schema/beans"`<br>    `xmlns:context="http://www.springframework.org/schema/context"` |

```xml
            xmlns:tx="http://www.springframework.org/schema/tx"
        xsi:schemaLocation="http://www.springframework.org/schema/mvc
http://www.springframework.org/schema/mvc/spring-mvc.xsd
                http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
                http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context.xsd
                http://www.springframework.org/schema/tx
http://www.springframework.org/schema/tx/spring-tx-4.1.xsd">


        <!-- DispatcherServlet Context: defines this servlet's request-processing
                infrastructure -->

        <!-- 어노테이션 설정 -->
        <context:annotation-config />

        <!-- Enables the Spring MVC @Controller programming model -->
        <annotation-driven />

        <!-- Handles HTTP GET requests for /resources/** by efficiently serving
                up static resources in the ${webappRoot}/resources directory -->
        <resources mapping="/resources/**" location="/resources/" />

        <!-- Resolves views selected for rendering by @Controllers to .jsp resources
                in the /WEB-INF/views directory -->
        <beans:bean
                class="org.springframework.web.servlet.view.InternalResourceViewResolver">
                <beans:property name="prefix" value="/WEB-INF/views/" />
                <beans:property name="suffix" value=".jsp" />
        </beans:bean>
        <!-- 경로 변경 -->
    <!-- <context:component-scan base-package="com.spring.dept" /> -->
        <context:component-scan base-package="com.spring.**" />
</beans:beans>
```

⑥ 스프링 설정 파일 생성 - 2

| webapp/WEB-INF/spring/**root-context.xml** |
| --- |
| ```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xmlns:context="http://www.springframework.org/schema/context"
``` |

```xml
        xsi:schemaLocation="http://www.springframework.org/schema/beans
http://www.springframework.org/schema/beans/spring-beans.xsd
                http://www.springframework.org/schema/context
http://www.springframework.org/schema/context/spring-context-4.2.xsd">


    <!-- Root Context: defines shared resources visible to all other web components -->


    <!-- DataSource 설정 -->
    <bean id="dataSource" class="org.apache.commons.dbcp.BasicDataSource" destroy-method="close">
        <property name="driverClassName" value="oracle.jdbc.OracleDriver" />
        <property name="url" value="jdbc:oracle:thin:@localhost:1521:orcl" />
        <property name="username" value="scott" />
        <property name="password" value="tiger" />
    </bean>


    <!-- Mybatis Setting -->
    <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
        <property name="dataSource" ref="dataSource"/>
        <property name="mapperLocations" value="classpath:mappings/*.xml"/>
    </bean>
    <bean id="sqlSession" class="org.mybatis.spring.SqlSessionTemplate"
        destroy-method="clearCache">
        <constructor-arg name="sqlSessionFactory" ref="sqlSessionFactory"></constructor-arg>
    </bean>


    <!-- a PlatformTransactionManager is still required -->
    <bean id="transactionManager"
class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
        <property name="dataSource" ref="dataSource" />
    </bean>
</beans>
```

- SpringMyBatisExam
  - src/main/java
    - com.spring.controller
      - DepartmentController.java
    - com.spring.dept
    - com.spring.deptdao
      - DeptMapper.java
      - DeptMapperImpl.java
    - com.spring.deptservice
      - DeptService.java
      - DeptServiceImpl.java
    - com.spring.deptvo
      - DeptVO.java
  - src/main/resources
    - mappings
      - dept-mapper.xml
    - META-INF
    - log4j.xml
  - src/test/java
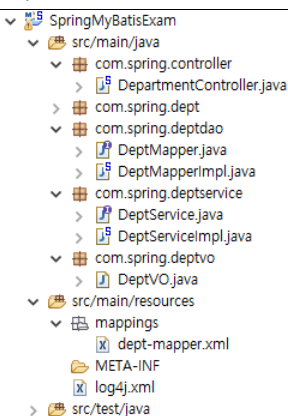
⑦ 컨트롤러, 서비스, DAO, 매퍼 XML 문서를 작성한다.

**DeptVO.java**

```java
package com.spring.deptvo;


public class DeptVO {
        private String    deptid="";      //학과코드
        private String    deptname=""; //학과명
        private String    depttel="";     //학과전화


        public String getDeptid() {
             return deptid;

        }
        public void setDeptid(String deptid) {
             this.deptid = deptid;

        }
        public String getDeptname() {
             return deptname;

        }
        public void setDeptname(String deptname) {
             this.deptname = deptname;

        }
        public String getDepttel() {
             return depttel;

        }
        public void setDepttel(String depttel) {
             this.depttel = depttel;

        }
}
```

**DepartmentController.java**

```java
package com.spring.deptcontroller;


import java.util.List;


import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestParam;
import org.springframework.web.servlet.ModelAndView;
```

```java
import com.spring.deptservice.DeptService;
import com.spring.deptvo.DeptVO;

/* 컨트롤러  */
@Controller
@RequestMapping(value="/department")
public class DepartmentController {
        private static final String CONTEXT_PATH="dept";

        @Autowired
        private DeptService deptService;

        /* 전체 조회 */
        @RequestMapping("/listDepartment")
        public ModelAndView listDepartment(@ModelAttribute DeptVO param) {
                List<DeptVO> list = deptService.listDepartment(param);

                ModelAndView mav = new ModelAndView();
                mav.addObject("departmentList", list);
                mav.setViewName(CONTEXT_PATH + "/department");

                return mav;
        }

        /* 상세 정보 보기 */
        @RequestMapping("/selectDepartment")
        public ModelAndView selectDepartment(@RequestParam("deptid") String deptid) {
            ModelAndView mav = new ModelAndView();

            if(deptid.equals("")) { // [등록] 버튼 클릭시
                mav.addObject("mode", "insert");
            } else {                //[수정/삭제] 버튼 클릭시
                mav.addObject("DeptVO", deptService.selectDepartment(deptid));
                mav.addObject("mode", "update");
            }
            mav.setViewName(CONTEXT_PATH + "/department_pop");

            return mav;
        }

        /* 레코드 추가 */
        @RequestMapping("/insertDepartment")
```

```java
public ModelAndView insertDepartment(@ModelAttribute DeptVO param) {
        String resultStr = "";
        int result = deptService.insertDepartment(param);

        if(result > 0) resultStr = "등록 완료이 완료되었습니다.";
        else resultStr = "등록에 문제가 있어 완료하지 못하였습니다.";

        ModelAndView mav = new ModelAndView();
        mav.addObject("result", resultStr);
        mav.setViewName("result");

        return mav;
}

/* 레코드 수정 */
@RequestMapping("/updateDepartment")
public ModelAndView updateDepartment(@ModelAttribute DeptVO param) {
        String resultStr = "";
        int result = deptService.updateDepartment(param);

        if(result > 0) resultStr = "수정 완료이 완료되었습니다.";
        else resultStr = "수정에 문제가 있어 완료하지 못하였습니다.";

        ModelAndView mav = new ModelAndView();
        mav.addObject("result", resultStr);
        mav.setViewName("result");

        return mav;
}

/* 레코드 삭제 */
@RequestMapping("/deleteDepartment")
public ModelAndView deleteDepartment(@ModelAttribute DeptVO param) {
        String resultStr = "";
        int result = deptService.deleteDepartment(param);

        if(result > 0) resultStr = "삭제 완료이 완료되었습니다.";
        else resultStr = "삭제에 문제가 있어 완료하지 못하였습니다.";

        ModelAndView mav = new ModelAndView();
        mav.addObject("result", resultStr);
        mav.setViewName("result");
```

```
                    return mav;
            }
}
```

## DeptService.java

```java
package com.spring.deptservice;

import java.util.List;
import com.spring.deptvo.DeptVO;

public interface DeptService {
    public List<DeptVO> listDepartment(DeptVO param);
    public DeptVO selectDepartment(String deptid);
    public int insertDepartment(DeptVO param);
    public int updateDepartment(DeptVO param);
    public int deleteDepartment(DeptVO param);
}
```

## DeptServiceImpl.java

```java
package com.spring.deptservice;

import java.util.ArrayList;
import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;
import com.spring.deptdao.DeptMapper;
import com.spring.deptvo.DeptVO;

@Service
@Transactional
public class DeptServiceImpl implements DeptService {

        @Autowired
        private DeptMapper deptMapper;

        @Override
        public List<DeptVO> listDepartment(DeptVO param) {
                List<DeptVO> list = new ArrayList<DeptVO>();
```

```java
                list = deptMapper.listDepartment(param);
                return list;
        }
        @Override
        public DeptVO selectDepartment(String deptid) {
                return deptMapper.selectDepartment(deptid);
        }
        @Override
        public int insertDepartment(DeptVO param) {
                return deptMapper.insertDepartment(param);
        }
        @Override
        public int updateDepartment(DeptVO param) {
                return deptMapper.updateDepartment(param);
        }
        @Override
        public int deleteDepartment(DeptVO param) {
                return deptMapper.deleteDepartment(param);
        }
}
```

**DeptMapper.java**

```java
package com.spring.deptdao;

import java.util.List;
import com.spring.deptvo.DeptVO;

public interface DeptMapper {
    public List<DeptVO> listDepartment(DeptVO param);
    public DeptVO selectDepartment(String deptid);
    public int insertDepartment(DeptVO param);
    public int updateDepartment(DeptVO param);
    public int deleteDepartment(DeptVO param);
}
```

**DeptMapperImpl.java**

```java
package com.spring.deptdao;

import java.util.List;
import javax.inject.Inject;
import org.apache.ibatis.session.SqlSession;
```

```java
import org.springframework.stereotype.Repository;
import com.spring.deptvo.DeptVO;

@Repository
public class DeptMapperImpl implements DeptMapper {
        @Inject
        private SqlSession sqlSession;

        private static final String namespace = "com.spring.deptdao.DeptMapper";

        @Override
        public List<DeptVO> listDepartment(DeptVO param) {
            return sqlSession.selectList(namespace+".listDepartment", param);
        }
        @Override
        public DeptVO selectDepartment(String deptid) {
            return sqlSession.selectOne(namespace+".selectDepartment", deptid);
        }
        @Override
        public int insertDepartment(DeptVO param) {
            return sqlSession.insert(namespace+".insertDepartment", param);
        }
        @Override
        public int updateDepartment(DeptVO param) {
            return sqlSession.update(namespace+".updateDepartment", param);
        }
        @Override
        public int deleteDepartment(DeptVO param) {
            return sqlSession.delete(namespace+".deleteDepartment", param);
        }
}
```

src/main/resources/mappings 폴더 생성

프로젝트명 -> New -> Other -> Java ORM Plugin ->Mybatis Mapper XML

| mappings/dept-mapper.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="com.spring.deptdao.DeptMapper">
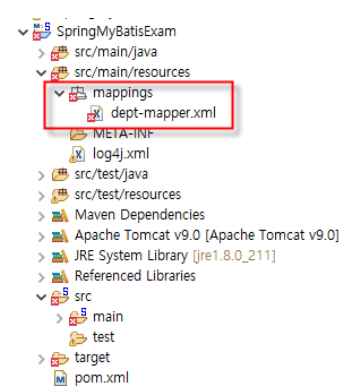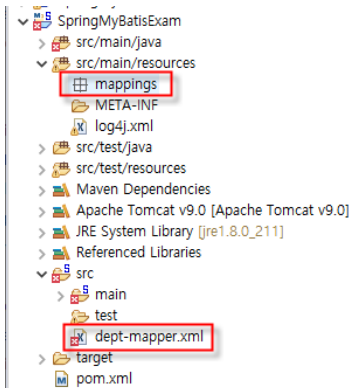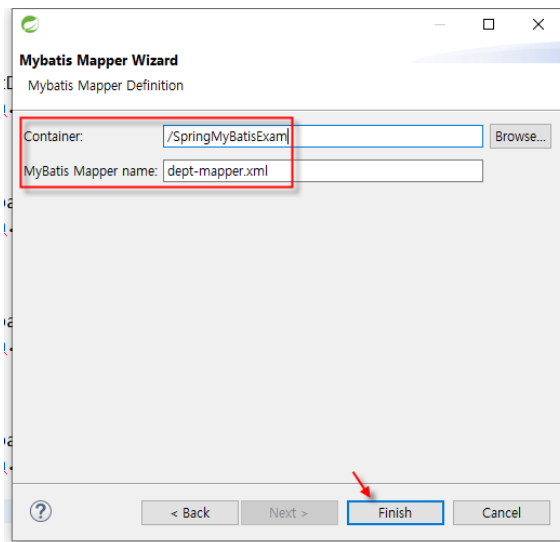        <resultMap type="com.spring.deptvo.DeptVO" id="deptVO">
                <result property="deptid"   column="Dept_ID" />
                <result property="deptname" column="Dept_Name" />
                <result property="depttel"   column="Dept_Tel" />
        </resultMap>

    <select id="listDepartment" parameterType="com.spring.deptvo.DeptVO"
resultMap="deptVO">
        /* Mapper - listDepartment */
                SELECT Dept_ID
                      ,Dept_NAME
                          ,Dept_Tel
                FROM    DEPARTMENT
                <where>
                    <if test="deptname != null and deptname != ''">
                        DEPT_NAME LIKE '%' || #{deptname} || '%'
                    </if>
                </where>
    </select>

    <select id="selectDepartment" parameterType="java.lang.String" resultMap="deptVO">
        /* Mapper - selectDepartment */
                SELECT Dept_ID
                      ,Dept_NAME
```

```
                            ,Dept_Tel
            FROM     DEPARTMENT
            WHERE    DEPT_ID      = #{deptid}
        </select>


    <insert id="insertDepartment" parameterType="com.spring.deptvo.DeptVO">
        /* Mapper - insertDepartment */
                INSERT INTO DEPARTMENT
                    ( DEPT_ID,    DEPT_NAME,   DEPT_TEL)
            VALUES
                    ( #{deptid}, #{deptname}, #{depttel} )
    </insert>


    <update id="updateDepartment" parameterType="com.spring.deptvo.DeptVO">
        /* Mapper - updateDepartment */
        UPDATE DEPARTMENT
        SET DEPT_NAME    = #{deptname},
            DEPT_TEL         = #{depttel}
        WHERE   DEPT_ID   = #{deptid}
    </update>


    <delete id="deleteDepartment" parameterType="com.spring.deptvo.DeptVO">
        /* Mapper - deleteDepartment */
        DELETE FROM DEPARTMENT   WHERE   DEPT_ID = #{deptid}
    </delete>
</mapper>
```

⑧ "views"에 "dept"폴더를 생성한다. 폴더를 선택하고 [New] - [JSP file]을 선택하고, 파일명으로 "department.jsp" 입력한 후, [Finish] 버튼을 클릭한다.
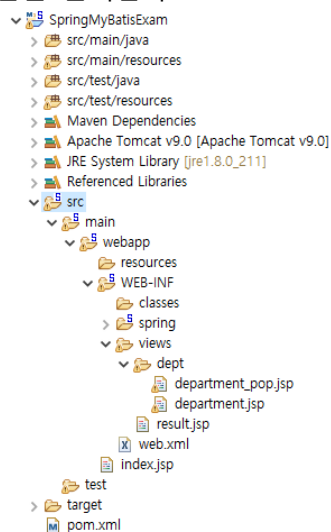
| **department.jsp** (/WEB-INF/views/dept/department.jsp) |
|---|

```jsp
<%@ page language="java" 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>학과 테이블</title>
<script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>
<script type="text/javascript">
        function insertPopup() {
                $("#deptid").val("");
                window.open("", "pop", "width=480, height=280");
                $("#popupForm").attr("action", "/department/selectDepartment.do");
                $("#popupForm").attr("target","pop");
                $("#popupForm").submit();
        }


        function updatePopup(deptid) {
                $("#deptid").val(deptid);
                window.open("", "pop", "width=480, height=280");
                $("#popupForm").attr("action", "/department/selectDepartment.do");
                $("#popupForm").attr("target","pop");
                $("#popupForm").submit();
        }


        function listSearch() {
                $("#searchForm").attr("action", "/department/listDepartment.do");
                $("#searchForm").submit();
        }
</script>
<style type="text/css">
        table{width:570px}
        #searchForm table th{width:114px}
        #searchForm table td{width:104px; text-align:center; }
        #deptname{width:328px}
</style>
</head>
<body>
        <form id="popupForm" name="popupForm" method="post" >
                <input type="hidden" name="deptid" id="deptid"/>
```

```
</form>
<div>
        <table border="0">
                <tr><th><h4>학과테이블[Spring]</h4></th></tr>
        </table>
</div>
<div>
        <table border="1">
                <colgroup>
                        <col width="120px" />
                        <col width="220px" />
                        <col width="120px" />
                        <col width="110px"/>
                </colgroup>
                <thead>
                        <tr>
                            <th>학과코드</th>
                            <th>학과명   </th>
                            <th>전화번호</th>
                            <th>상세보기</th>
                        </tr>
                </thead>
                <tbody>
                        <c:if test="${empty departmentList}">
                                <tr>
                                    <td colspan="5" align="center">
                                        등록된 학과 정보가 존재하지 않습니다.</td>
                                </tr>
                        </c:if>
                        <c:forEach items="${departmentList}" var="row">
                                <tr>
                                    <td align="center" >${row.deptid}</td>
                                    <td align="left"    >${row.deptname}</td>
                                    <td align="center" >${row.depttel}</td>
                                    <td align="center" >
                                        <input type="button" value="[ 수정/삭제  ]"
                                         onclick="updatePopup('${row.deptid}')"/></td>
                                </tr>
                        </c:forEach>
                        <tr>
                            <td colspan="3" align="center">
                                처리할 버튼을 선택하세요.</td>
```

```
                                            <td align="center">
                                                <input type="button" onclick="insertPopup()"
                                                    value="[ 등록 ]"/>
                                            </td>
                                        </tr>
                                    </tbody>
                                </table>
                        </div>
                        <p></p>
                        <div><!-- search -->
                                <form id="searchForm" name="searchForm" method="post">
                                        <table border="1" cellpadding="1" cellspacing="1">
                                                <tr>
                                                        <th>학 과 명</th>
                                                        <td><input type="text" id="deptname"
                                                                name="deptname" /></td>
                                                        <td><input type="button" onclick="listSearch()"
                                                                value="검색" /></td>
                                                </tr>
                                        </table>
                                </form>
                        </div>
        </body>
        </html>
```

[New] - [JSP file]을 선택하고, 파일명으로 "department_pop.jsp" 입력한 후, [Finish] 버튼을 클릭한다.

**department_pop.jsp** (/WEB-INF/views/dept/department_pop.jsp)

```
<%@ page language="java" 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>학과 테이블 팝업</title>
<style type="text/css">
        .required{color:red;}
        table{width:450px;}
        table th{width:120px; text-align:left;}
        input[type="text"]{width:330px;}
</style>
```

```html
<script type="text/javascript" src="http://code.jquery.com/jquery-latest.js"></script>
<script type="text/javascript">
        $(document).ready(function(){
                var mode = "${mode}";
                if(mode == "insert") {
                        $('#insertData').attr('disabled',false);
                        $('#updateData').attr('disabled',true);
                        $('#deleteData').attr('disabled',true);
                } else if(mode == "update") {
                        $('#insertData').attr('disabled',true);
                        $('#updateData').attr('disabled',false);
                        $('#deleteData').attr('disabled',false);
                }

                $("#insertData").click(function(){
                        if(!validateForm()) return;
                        if(confirm('등록을 진행할까요?')) {
                                $("#deptForm").attr("action", "/department/insertDepartment.do");
                                $("#deptForm").submit();
                        }
                });

                $("#updateData").click(function(){
                        if(!validateForm()) return;
                        if(confirm('수정을 진행할까요?')) {
                                $("#deptForm").attr("action", "/department/updateDepartment.do");
                                $("#deptForm").submit();
                        }
                });

                $("#deleteData").click(function(){
                        if(confirm('삭제를 진행할까요?')) {
                                $("#deptForm").attr("action", "/department/deleteDepartment.do");
                                $("#deptForm").submit();
                        }
                });

                $("#closeWindow").click(function(){
                        window.close();
                });
        });
```

```
        function validateForm() {
            if($("#deptid").val().replace(/₩s/g,"") == "") {
                alert('학과코드를 입력해주세요.');
                return false;
            }
            if($("#deptname").val().replace(/₩s/g,"") == "") {
                alert('학과명을 입력해주세요.');
                return false;
            }
            return true;
        }
</script>
</head>
<body>
        <p></p>
        <div>
                <form id="deptForm" name="deptForm" method="post">
                        <table border="1">
                                <thead>
                                        <tr>
                                            <td colspan="2" align="center">
                                                <h4>학과 테이블 팝업창 [Spring] </h4></td>
                                        </tr>
                                </thead>
                                <tbody>
                                        <tr>
                                            <th><span class="required">*</span>학과코드</th>
                                            <td><input type="text" id="deptid"   name="deptid"
                                                    value="${DeptVO.deptid}" /></td>
                                        </tr>
                                        <tr>
                                            <th><span class="required">*</span>학과명</th>
                                            <td><input type="text" id="deptname"
                                                name="deptname" value="${DeptVO.deptname}" />
                                            </td>
                                        </tr>
                                        <tr>
                                            <th><span class="required">*</span>전화번호</th>
                                            <td><input type="text" id="depttel" name="depttel"
                                                    value="${DeptVO.depttel}" /></td>
                                        </tr>
                                </tbody>
```

```
                           </table>
                   </form>
           </div>
           <div>

                   <p></p>
                   <table border="0">
                           <tr align="center">
                               <td>
                                       <input type="button" id="insertData" value="등록" /></td>
                               <td>
                                       <input type="button" id="updateData" value="수정" /></td>
                               <td>
                                       <input type="button" id="deleteData" value="삭제" /></td>
                               <td>
                                       <input type="button" id="closeWindow" value="닫기" /></td>
                           </tr>
                   </table>
           </div>
</body>
</html>
```

| result.jsp (/WEB-INF/views/result.jsp) |
| --- |

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>결과 화면</title>
<script type="text/javascript">
        alert("${result}");
        if("${result}".indexOf("문제") > -1) {
            history.go(-1);
        } else {
            opener.listSearch();
            window.close();
        }
</script>
</head>
<body></body>
</html>
```

| index.jsp(/webapp/index.jsp) |
|---|
| <%@ page language="java" contentType="text/html; charset=UTF-8"<br>    pageEncoding="UTF-8"%><br><!DOCTYPE html><br><html><br><head><br><meta charset="UTF-8"><br><title>스프링 MVC 데이터베이스 연동</title><br><style type="text/css"><br>        a{text-decoration:none;}<br></style><br></head><br><body><br>        <a href="department/listDepartment.do">[학과 데이터]</a><br></body><br></html> |

⑨ index.jsp 실행하고 난 후 톰캣을 멈추고 Server 탭의 Tomcat v9.0..을 더블 클릭한다. Modules 탭에서 다음과 같이 수정하고 저장한다.(또는 Add Web Module.. 버튼을 클릭하여 Web Modules 에 Path와 Document Base에 /dept 와 /SpringMyBatisExam 가 나타나도록 선택한다.)
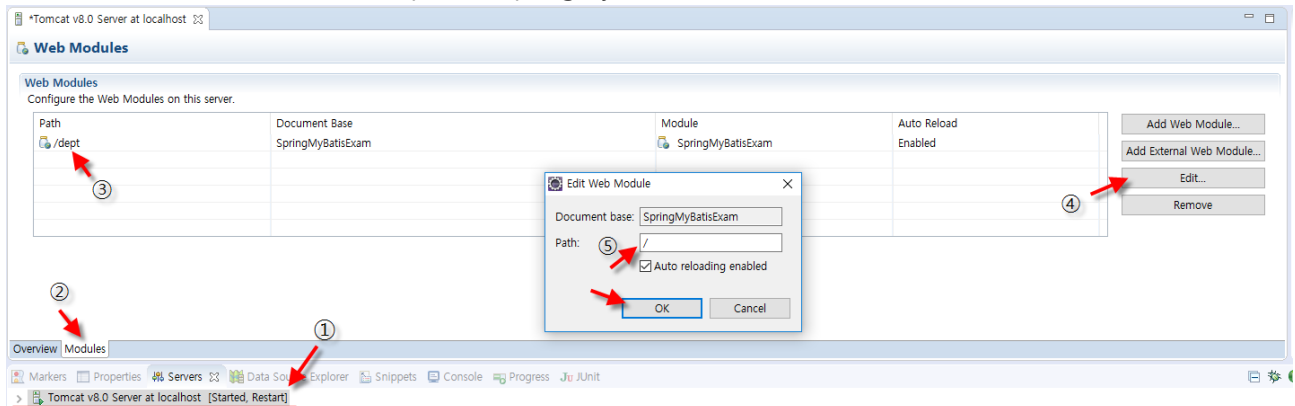


위 내용을 설정하면 [Servers] - [server.xml] 파일에 다음과 같은 소스가 자동으로 추가된다.

```
<Context docBase="SpringMyBatisExam" path="/" reloadable="true"
    source="org.eclipse.jst.jee.server:SpringMyBatisExam"/>
```

수정 후 실행해 본다.



207