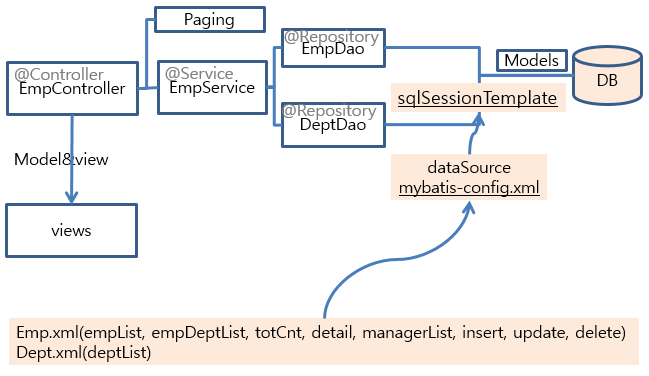
[ 15 ] Mybatis - II

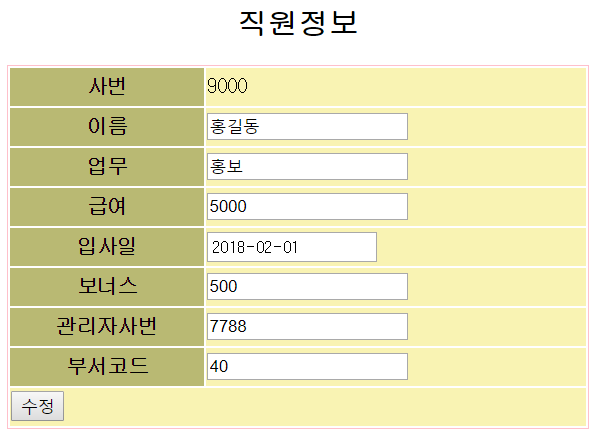
처음 JSP&Servlet을 이용할 때는 많은 코드양으로 DB에 접근하다가, Spring에서는 JDBC Template을 이용해서 코드양을 많이 줄일 수 있었다. Mybatis를 이용하면 코드를 더 줄일 수 있고, 자바 코드가 아닌 xml 파일로 DB에 엑세스한다. MyBatis는 DB접근을 자바가 아닌 xml로 대체하도록 하는 라이브러리이다. SQL 코드를 자바코드와 분리시키는데 목적이 있다.











**pom.xml**

<!-- lombok -->

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<version>1.18.26</version>

<scope>provided</scope>

</dependency>

<!-- JDBC template -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>4.1.4.RELEASE</version>

</dependency>

<!-- myBatis -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>3.2.8</version>

</dependency>

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis-spring</artifactId>

<version>1.2.2</version>

</dependency>

**web.xml**

<servlet-mapping>

<servlet-name>appServlet</servlet-name>

<url-pattern>\*.do</url-pattern>

</servlet-mapping>

<!-- 한글처리 -->

<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>

**index.jsp**

<body>

<jsp:forward page=*"home.do"*/>

<%-- <jsp:forward page="list.do"/> --%>

<%-- <jsp:forward page="joinList.do"/> --%>

<%-- <jsp:forward page="insert50.do"/> --%>

</body>

**HomeController.java**

**import** org.springframework.stereotype.Controller;

**import** org.springframework.web.bind.annotation.RequestMapping;

**import** org.springframework.web.bind.annotation.RequestMethod;

@Controller

**public** **class** HomeController {

@RequestMapping(value = "home", method = RequestMethod.***GET***)

**public** String home() {

**return** "home";

}

}

**src/main/resources/META-INF/property/db.properties**

db.driverClassName=oracle.jdbc.driver.OracleDriver

db.url=jdbc:oracle:thin:@localhost:1521:xe

db.username=teamname

db.password=teamname

create user teamname identified by teampw;

grant dba to teamname;

@c:/demo.sql

**src/main/resources/META-INF/db/ch16emp.sql**

-- Dept.xml의 id=deptList

SELECT \* FROM DEPT;

-- Emp.xml의 id=empList

SELECT \* FROM EMP ORDER BY EMPNO DESC;

SELECT ROWNUM RN, A.\* FROM (SELECT \* FROM EMP ORDER BY EMPNO DESC) A;

SELECT \* FROM

(SELECT ROWNUM RN, A.\* FROM (SELECT \* FROM EMP ORDER BY EMPNO DESC) A)

WHERE RN BETWEEN 4 AND 6;

-- Emp.xml의 id=total

SELECT COUNT(\*) FROM EMP;

-- Emp.xml의 id=detail

SELECT \* FROM EMP WHERE EMPNO=7863;

-- Emp.xml의 id=managerList

SELECT \* FROM EMP WHERE EMPNO IN (SELECT MGR FROM EMP);

-- Emp.xml의 id=insert

INSERT INTO EMP VALUES (9000, 'HONG', 'IT',7566,'18/02/01',8000,200,40);

-- Emp.xml의 id=update

UPDATE EMP SET ENAME='홍', JOB='개발', MGR=7566,HIREDATE='2018-02-01',SAL=9000, COMM=900, DEPTNO=30

WHERE EMPNO=9000;

SELECT \* FROM EMP;

ROLLBACK;

-- Emp.xml의 id=delete

DELETE FROM EMP WHERE EMPNO=9000;

-- EmpDept.xml의 id=empDeptList

SELECT \* FROM

(SELECT ROWNUM RN, A.\* FROM (SELECT E.\*, DNAME, LOC FROM EMP E, DEPT D

WHERE E.DEPTNO=D.DEPTNO ORDER BY EMPNO DESC) A)

WHERE RN BETWEEN 4 AND 6;

**src/main/java/com/lec/ch15/model/Dept.java**

**import** lombok.Data;

@Data

**public** **class** Dept {

**private** **int** deptno;

**private** String dname;

**private** String loc;

}

**src/main/java/com/lec/ch15/model/Emp.java**

**import** java.sql.Date;

**import** lombok.Data;

@Data

**public** **class** Emp {

**private** **int** empno;

**private** String ename;

**private** String job;

**private** **int** mgr;

**private** Date hiredate;

**private** **int** sal;

**private** **int** comm;

**private** **int** deptno;

**private** String dname;

**private** String loc;

**private** **int** startRow; // 페이징 처리를 위한 속성 변수

**private** **int** endRow; // 페이징 처리를 위한 속성 변수

}

**src/main/resources/META-INF/sqlmap/config/mybatis-config.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>

<typeAliases>

<typeAlias type=*"com.lec.ch15.model.Dept"* alias=*"Dept"*/>

<typeAlias type=*"com.lec.ch15.model.Emp"* alias=*"Emp"*/>

</typeAliases>

<mappers>

<mapper resource=*"META-INF/sqlmap/mappers/Dept.xml"*/>

<mapper resource=*"META-INF/sqlmap/mappers/Emp.xml"*/>

</mappers>

</configuration>

**src/main/resources/META-INF/sqlmap/mappers/Dept.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=*"Dept"*>

<resultMap type=*"Dept"* id=*"DeptResult"*>

<result property=*"deptno"* column=*"deptno"*/>

<result property=*"dname"* column=*"dname"*/>

<result property=*"loc"* column=*"loc"*/>

</resultMap>

<select id=*"deptList"* resultMap=*"DeptResult"*>

<![CDATA[

SELECT \* FROM DEPT

]]>

</select>

</mapper>

**src/main/resources/META-INF/sqlmap/mappers/Emp.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=*"Emp"*>

<resultMap type=*"Emp"* id=*"EmpResult"*>

<result property=*"empno"* column=*"empno"*/>

<result property=*"ename"* column=*"ename"*/>

<result property=*"job"* column=*"job"*/>

<result property=*"mgr"* column=*"mgr"*/>

<result property=*"hiredate"* column=*"hiredate"*/>

<result property=*"sal"* column=*"sal"*/>

<result property=*"comm"* column=*"comm"*/>

<result property=*"deptno"* column=*"deptno"*/>

<result property=*"dname"* column=*"dname"*/>

<result property=*"loc"* column=*"loc"*/>

</resultMap>

<select id=*"empList"* parameterType=*"Emp"* resultMap=*"EmpResult"*>

<![CDATA[

SELECT \* FROM (SELECT ROWNUM RN, A.\* FROM (SELECT \* FROM EMP ORDER BY EMPNO) A)

WHERE RN BETWEEN #{startRow} AND #{endRow}

]]>

</select>

<select id=*"empDeptList"* parameterType=*"Emp"* resultMap=*"EmpResult"*>

SELECT \* FROM (SELECT ROWNUM RN, A.\* FROM

(SELECT E.\*, DNAME, LOC FROM EMP E, DEPT D

WHERE E.DEPTNO=D.DEPTNO ORDER BY EMPNO) A)

WHERE RN BETWEEN #{startRow} AND #{endRow}

</select>

<select id=*"totCnt"* resultType=*"int"*>

SELECT COUNT(\*) FROM EMP

</select>

<select id=*"detail"* parameterType=*"int"* resultType=*"Emp"*>

SELECT \* FROM EMP WHERE EMPNO=#{empno, jdbcType=VARCHAR}

</select>

<select id=*"managerList"* resultMap=*"EmpResult"*>

SELECT \* FROM EMP WHERE EMPNO IN (SELECT MGR FROM EMP)

</select>

<insert id=*"insert"* parameterType=*"Emp"*>

INSERT INTO EMP VALUES (#{empno}, #{ename},#{job}, #{mgr}, #{hiredate}, #{sal}, #{comm}, #{deptno})

</insert>

<update id=*"update"* parameterType=*"Emp"*>

UPDATE EMP SET ENAME=#{ename},

JOB =#{job},

MGR=#{mgr},

HIREDATE=#{hiredate},

SAL = #{sal},

COMM = #{comm},

DEPTNO=#{deptno}

WHERE EMPNO=#{empno}

</update>

<delete id=*"delete"* parameterType=*"int"*>

DELETE FROM EMP WHERE EMPNO=#{empno}

</delete>

</mapper>

**servlet-context.xml**

<resources mapping=*"/css/\*\*"* location=*"/css/"* />

**root-context.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"*

xsi:schemaLocation=*"http://www.springframework.org/schema/beans https://www.springframework.org/schema/beans/spring-beans.xsd*

*http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-3.1.xsd"*>

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

<context:property-placeholder

location=*"classpath:META-INF/property/db.properties"*/>

<!-- DataSource 빈 -->

<bean id=*"dataSource"* class=*"org.springframework.jdbc.datasource.DriverManagerDataSource"*>

<property name=*"driverClassName"* value=*"${db.driverClassName}"*/>

<property name=*"url"* value=*"${db.url}"*/>

<property name=*"username"* value=*"${db.username}"*/>

<property name=*"password"* value=*"${db.password}"*/>

</bean>

<!-- SqlSessionFactoryBean 빈 -->

<bean id=*"sqlSessionFactory"* class=*"org.mybatis.spring.SqlSessionFactoryBean"*>

<property name=*"dataSource"* ref=*"dataSource"*/>

<property name=*"configLocation"* value=*"classpath:META-INF/sqlmap/config/mybatis-config.xml"*/>

</bean>

<!-- SqlSessionTemplate 빈 필요 -->

<bean id=*"sessionTemplate"* class=*"org.mybatis.spring.SqlSessionTemplate"*>

<constructor-arg ref=*"sqlSessionFactory"*/>

</bean>

</beans>

**src/main/java/com/lec/ch15/dao/DeptDao.java**

**public** **interface** DeptDao {

**public** List<Dept> deptList();

}

**src/main/java/com/lec/ch15/dao/DeptDaoImpl.java**

**import** java.util.List;

**import** org.mybatis.spring.SqlSessionTemplate;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Repository;

**import** com.lec.ch15.model.Dept;

@Repository // 스프링에게 DAO임을 알려주는 어노테이션

**public** **class** DeptDaoImpl **implements** DeptDao {

@Autowired

**private** SqlSessionTemplate sessionTemplate;

@Override

**public** List<Dept> deptList() {

**return** sessionTemplate.selectList("deptList");

}

}

**src/main/java/com/lec/ch15/dao/EmpDao.java**

**import** java.util.List;

**import** com.lec.ch15.model.Emp;

**public** **interface** EmpDao {

**public** List<Emp> empList(Emp emp);

**public** List<Emp> empDeptList(Emp emp);

**public** **int** totCnt();

**public** Emp detail(**int** empno);

**public** List<Emp> managerList();

**public** **int** insert(Emp emp);

**public** **int** update(Emp emp);

**public** **int** delete(**int** empno);

}

**src/main/java/com/lec/ch15/dao/EmpDaoImpl.java**

**import** java.util.List;

**import** org.apache.ibatis.session.SqlSession;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Repository;

**import** com.lec.ch15.model.Emp;

@Repository

**public** **class** EmpDaoImpl **implements** EmpDao{

@Autowired

**private** SqlSession sessionTemplate;

@Override

**public** List<Emp> empList(Emp emp) {

**return** sessionTemplate.selectList("empList", emp);

}

@Override

**public** List<Emp> empDeptList(Emp emp) {

**return** sessionTemplate.selectList("empDeptList", emp);

}

@Override

**public** **int** totCnt() {

**return** sessionTemplate.selectOne("totCnt");

}

@Override

**public** Emp detail(**int** empno) {

**return** sessionTemplate.selectOne("detail", empno);

}

@Override

**public** List<Emp> managerList() {

**return** sessionTemplate.selectList("managerList");

}

@Override

**public** **int** insert(Emp emp) {

**return** sessionTemplate.insert("insert", emp);

}

@Override

**public** **int** update(Emp emp) {

**return** sessionTemplate.update("update", emp);

}

@Override

**public** **int** delete(**int** empno) {

**return** sessionTemplate.delete("delete", empno);

}

}

**src/main/java/com/lec/ch15/service/EmpService.java**

**public** **interface** EmpService {

**public** List<Dept> deptList();

**public** List<Emp> empList(String pageNum);

**public** List<Emp> empDeptList(String pageNum);// paging-service단

**public** **int** totCnt();

**public** Emp detail(**int** empno);

**public** List<Emp> managerList();

**public** **int** insert(Emp emp);

**public** **int** update(Emp emp);

**public** **int** delete(**int** empno);

**public** **void** dummyDataInsert50();

}

**src/main/java/com/lec/ch15/service/Paging.java**

**import** lombok.Data;

@Data

**public** **class** Paging {

**private** **int** currentPage = 1;

**private** **int** pageSize;

**private** **int** blockSize;

**private** **int** startRow;

**private** **int** endRow;

**private** **int** totCnt;

**private** **int** pageCnt;

**private** **int** startPage;

**private** **int** endPage;

// Paing paging = new Paging(14, pageNum, 10, 10)

**public** Paging(**int** totCnt, String pageNum, **int** pageSize, **int** blockSize) {

**if**(pageNum!=**null**) {

currentPage = Integer.*parseInt*(pageNum);

}

**this**.pageSize = pageSize;

**this**.blockSize = blockSize;

startRow = (currentPage-1) \* pageSize +1;

endRow = startRow + pageSize -1;

**this**.totCnt = totCnt;

pageCnt = (**int**)Math.*ceil*((**double**)totCnt/pageSize);

startPage = ((currentPage-1) / blockSize) \* blockSize +1;

startPage = currentPage - (currentPage-1)%blockSize;

endPage = startPage + blockSize -1;

**if**(endPage > pageCnt) {

endPage = pageCnt;

}

}

}

**src/main/java/com/lec/ch15/service/EmpServiceImpl.java**

**import** java.sql.Date;

**import** java.util.List;

**import** org.springframework.beans.factory.annotation.Autowired;

**import** org.springframework.stereotype.Service;

**import** com.lec.ch15.dao.DeptDao;

**import** com.lec.ch15.dao.EmpDao;

**import** com.lec.ch15.model.Dept;

**import** com.lec.ch15.model.Emp;

**import** com.lec.ch15.util.Paging;

@Service

**public** **class** EmpServiceImpl **implements** EmpService {

@Autowired

**private** EmpDao empDao;

@Autowired

**private** DeptDao deptDao;

@Override

**public** List<Dept> deptList() {

**return** deptDao.deptList();

}

@Override

**public** List<Emp> empList(String pageNum) { // startRow, endRow는 controller에서

Paging paging = **new** Paging(empDao.totCnt(), pageNum, 10, 10);

Emp emp = **new** Emp();

emp.setStartRow(paging.getStartRow());

emp.setEndRow(paging.getEndRow());

**return** empDao.empList(emp);

}

@Override

**public** List<Emp> empDeptList(String pageNum) {

Paging paging = **new** Paging(empDao.totCnt(), pageNum, 10, 10);

Emp emp = **new** Emp();

emp.setStartRow(paging.getStartRow());

emp.setEndRow(paging.getEndRow());

**return** empDao.empDeptList(emp);

}

@Override

**public** **int** totCnt() {

**return** empDao.totCnt();

}

@Override

**public** Emp detail(**int** empno) {

**return** empDao.detail(empno);

}

@Override

**public** List<Emp> managerList() {

**return** empDao.managerList();

}

@Override

**public** **int** insert(Emp emp) {

**return** empDao.insert(emp);

}

@Override

**public** **int** update(Emp emp) {

**return** empDao.update(emp);

}

@Override

**public** **int** delete(**int** empno) {

**return** empDao.delete(empno);

}

@Override

**public** **void** dummyDataInsert50() {

Emp emp = **new** Emp();

**for**(**int** i=3000; i<3050 ; i++) {

emp.setEmpno(i);

emp.setEname("홍"+i);

emp.setJob("IT");

emp.setMgr(7566);

//emp.setHiredate(Timestamp.valueOf("2022-07-21 15:43:00"));

emp.setHiredate(Date.*valueOf*("2022-07-21"));

emp.setSal(i);

emp.setComm(i/10);

emp.setDeptno(40);

**int** result = empDao.insert(emp);

System.***out***.println(result==1? i+"번째 성공":"실패");

}

}

}

**src/main/java/com/ch/ch16/controller/EmpController.java**

@Controller

**public** **class** EmpController {

@Autowired

**private** EmpService empService;

@RequestMapping(value="empList", method = RequestMethod.***GET***)

**public** String empList(String pageNum, Model model) {

// empList.do?pageNum=2, empList.do

model.addAttribute("empList", empService.empList(pageNum));

model.addAttribute("paging", **new** Paging(empService.totCnt(), pageNum, 10, 5));

**return** "empList";

}

@RequestMapping(value="dummyDataInsert", method = RequestMethod.***GET***)

**public** String dummyDataInsert() {

empService.dummyDataInsert50();

**return** "redirect:empList.do";

}

@RequestMapping(value="empDeptList", method= {RequestMethod.***GET***, RequestMethod.***POST***})

**public** String empDeptList(String pageNum, Model model) {

//empDeptList.do , empDeptList.do?pageNum=2

model.addAttribute("empList", empService.empDeptList(pageNum));

model.addAttribute("paging", **new** Paging(empService.totCnt(), pageNum, 10, 5));

**return** "empDeptList";

}

@RequestMapping(value="detail", method = RequestMethod.***GET***)

**public** String detail(**int** empno, Model model) {

model.addAttribute("empDto", empService.detail(empno));

**return** "detail";

}

@RequestMapping(value="updateView", method = {RequestMethod.***GET***, RequestMethod.***POST***})

**public** String updateView(**int** empno, Model model) {

model.addAttribute("empDto", empService.detail(empno));

**return** "update";

}

@RequestMapping(value="update", method = RequestMethod.***POST***)

**public** String update(Emp emp, Model model) {

**try** {

model.addAttribute("updateResult", empService.update(emp));

}**catch**(Exception e) {

model.addAttribute("updateResult", "필드 값이 너무 깁니다. 수정 실패");

**return** "forward:updateView.do";

}

**return** "forward:empDeptList.do";

}

@RequestMapping(value = "delete", method = RequestMethod.***GET***)

**public** String delete(**int** empno, Model model) {

model.addAttribute("deleteResult", empService.delete(empno));

**return** "forward:empDeptList.do";

}

@RequestMapping(value = "writeView", method = {RequestMethod.***GET***, RequestMethod.***POST***})

**public** String writeView(Model model) {

model.addAttribute("managerList", empService.managerList());

model.addAttribute("deptList", empService.deptList());

**return** "write";

}

@RequestMapping(value = "confirmNo", method = RequestMethod.***GET***)

**public** String confirmNo(Emp emp, Model model) {

**if**(empService.detail(emp.getEmpno()) == **null**) {

model.addAttribute("msg", "사용가능한 사번입니다");

}**else** {

model.addAttribute("msg", "중복된 사번은 사용불가합니다");

}

**return** "forward:writeView.do";

}

@RequestMapping(value = "write", method = RequestMethod.***POST***)

**public** String write(Emp emp, Model model) {

**try** {

model.addAttribute("writeResult", empService.insert(emp));

}**catch** (Exception e) {

model.addAttribute("writeResult", "필드값이 너무 길어 등록 불가");

**return** "forward:writeView.do";

}

**return** "forward:empDeptList.do";

}

}

**board.css**

@charset *"UTF-8"*;

**body** { font-family: *굴림체*; text-align: *center*; }

**table** { border: *1px* *solid* *pink*; width: *100%*; }

**tr** { height: *30px*; background: *yellow*; }

**th** { background: *#b9b973*; }

**td** { background: *#f9f3b3*; }

**caption** {font-size: *2em*; }

**b**{color:*red*;}

**a**{text-decoration: *none*;}

**empList.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<%@ taglib prefix=*"fmt"* uri=*"http://java.sun.com/jsp/jstl/fmt"*%>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*${conPath }*/css/style.css"* rel=*"stylesheet"*>

<script src=*"https://code.jquery.com/jquery-3.6.0.js"*></script>

<script>

$(document).ready(**function**(){

// $('tr').click(function(){

// var empno = Number($(this).children().eq(1).html());

// if(!isNaN(empno)){

// location.href = '${conPath}/detail.do?empno='+empno + '&pageNum=${paging.currentPage}';

// }

// });

});

**function** trClicked(empno) {

location.href = '${conPath}/detail.do?empno=' + empno + '&pageNum=${paging.currentPage}';

}

</script>

</head>

<body>

<c:set var=*"num"* value=*"*${paging.totCnt - paging.startRow + 1 }*"*/>

<table>

<caption>직원 목록</caption>

<tr>

<td colspan=*"6"* align=*"right"*><a href=*"*${conPath }*/write.do"*>직원등록</a></td>

</tr>

<tr><th>번호</th><th>사번</th><th>이름</th><th>급여</th><th>입사일</th><th>부서번호</th></tr>

<c:forEach var=*"emp"* items=*"*${empList }*"*>

<tr onclick=*"trClicked('*${emp.empno}*')"*>

<td>${num }</td>

<td>${emp.empno }</td>

<td>${emp.ename }</td>

<td><fmt:formatNumber value=*"*${emp.sal}*"* groupingUsed=*"true"*/></td>

<td>${emp.hiredate }</td>

<td>${emp.deptno }</td>

</tr>

<c:set var=*"num"* value=*"*${num-1 }*"* />

</c:forEach>

</table>

<br>

<br>

<div class=*"paging"*>

<c:if test=*"*${paging.startPage > paging.blockSize }*"*>

[ <a href=*"*${conPath }*/empList.do?pageNum=*${paging.startPage-1}*"*>이전</a> ]

</c:if>

<c:forEach var=*"i"* begin=*"*${paging.startPage }*"*

end=*"*${paging.endPage }*"*>

<c:if test=*"*${paging.currentPage **eq** i }*"*>

[ <b> ${i }</b> ]

</c:if>

<c:if test=*"*${paging.currentPage!=i }*"*>

[ <a href=*"*${conPath }*/empList..do?pageNum=*${i}*"*> ${i }</a> ]

</c:if>

</c:forEach>

<c:if test=*"*${paging.endPage<paging.pageCnt }*"*>

[ <a href=*"*${conPath }*/empList..do?pageNum=*${paging.endPage+1}*"*>다음</a> ]

</c:if>

</div>

</body>

</html>

**empDeptList.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<%@ taglib prefix=*"fmt"* uri=*"http://java.sun.com/jsp/jstl/fmt"*%>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*${conPath }*/css/style.css"* rel=*"stylesheet"*>

<script src=*"https://code.jquery.com/jquery-3.6.0.js"*></script>

<script>

$(document).ready(**function**(){

// $('tr').click(function(){

// var empno = Number($(this).children().eq(1).html());

// if(!isNaN(empno)){

// location.href = '${conPath}/detail.do?empno='+empno + '&pageNum=${paging.currentPage}';

// }

// });

});

**function** trClicked(empno) {

location.href = '${conPath}/detail.do?empno=' + empno + '&pageNum=${paging.currentPage}';

}

</script>

</head>

<body>

<c:if test=*"*${updateResult **eq** 1 }*"*>

<script>alert('직원 정보 수정 성공');</script>

</c:if>

<c:if test=*"*${deleteResult **eq** 1 }*"*>

<script>alert('직원 삭제 성공');</script>

</c:if>

<c:if test=*"*${writeResult **eq** 1 }*"*>

<script>alert('직원 등록 성공');</script>

</c:if>

<c:set var=*"num"* value=*"*${paging.totCnt - paging.startRow + 1 }*"*/>

<table>

<caption>직원 목록</caption>

<tr>

<td colspan=*"6"* align=*"right"*><a href=*"*${conPath }*/writeForm.do"*>직원등록</a></td>

</tr>

<tr><th>번호</th><th>사번</th><th>이름</th><th>급여</th><th>부서이름</th><th>근무지</th></tr>

<c:forEach var=*"emp"* items=*"*${empList }*"*>

<tr onclick=*"trClicked('*${emp.empno}*')"*>

<td>${num }</td>

<td>${emp.empno }</td>

<td>${emp.ename }</td>

<td><fmt:formatNumber value=*"*${emp.sal}*"* groupingUsed=*"true"*/></td>

<td>${emp.dname }</td>

<td>${emp.loc }</td>

</tr>

<c:set var=*"num"* value=*"*${num-1 }*"* />

</c:forEach>

</table>

<br>

<br>

<div class=*"paging"*>

<c:if test=*"*${paging.startPage > paging.blockSize }*"*>

[ <a href=*"*${conPath }*/empDeptList.do?pageNum=*${paging.startPage-1}*"*>이전</a> ]

</c:if>

<c:forEach var=*"i"* begin=*"*${paging.startPage }*"*

end=*"*${paging.endPage }*"*>

<c:if test=*"*${paging.currentPage==i }*"*>

[ <b> ${i }</b> ]

</c:if>

<c:if test=*"*${paging.currentPage!=i }*"*>

[ <a href=*"*${conPath }*/empDeptList.do?pageNum=*${i}*"*> ${i }</a> ]

</c:if>

</c:forEach>

<c:if test=*"*${paging.endPage<paging.pageCnt }*"*>

[ <a href=*"*${conPath }*/empDeptList.do?pageNum=*${paging.endPage+1}*"*>다음</a> ]

</c:if>

</div>

</body>

</html>

**detail.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<%@ taglib prefix=*"fmt"* uri=*"http://java.sun.com/jsp/jstl/fmt"*%>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*${conPath }*/css/style.css"* rel=*"stylesheet"*>

<script src=*"https://code.jquery.com/jquery-3.6.0.js"*></script>

<script>

$(document).ready(**function**(){

});

</script>

</head>

<body>

<table>

<caption>직원정보</caption>

<tr><th>사번</th><td>${empDto.empno }</td></tr>

<tr><th>이름</th><td>${empDto.ename }</td></tr>

<tr><th>업무</th><td>${empDto.job }</td></tr>

<tr><th>관리자번호</th><td>${empDto.mgr }</td></tr>

<tr><th>입사일</th><td><fmt:formatDate value=*"*${empDto.hiredate }*"* dateStyle=*"long"*/> </td></tr>

<tr><th>급여</th><td>${empDto.sal }</td></tr>

<tr><th>상여</th><td>${empDto.comm }</td></tr>

<tr><th>부서번호</th><td>${empDto.deptno }</td></tr>

<tr><td colspan=*"2"* align=*"center"*>

<button onclick=*"location.href='empDeptList.do?pageNum=*${param.pageNum}*'"*>목록</button>

<button onclick=*"location.href='update.do?empno=*${empDto.empno }*&pageNum=*${param.pageNum}*'"*>수정</button>

<button onclick=*"location.href='delete.do?empno=*${empDto.empno }*&pageNum=*${param.pageNum}*'"*>삭제</button>

</td></tr>

</table>

</body>

</html>

**Write.jsp**

<%@page import=*"java.sql.Date"*%>

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<%@ taglib prefix=*"fmt"* uri=*"http://java.sun.com/jsp/jstl/fmt"*%>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*${conPath }*/css/style.css"* rel=*"stylesheet"*>

<script src=*"https://code.jquery.com/jquery-3.6.0.js"*></script>

<script>

$(document).ready(**function**(){

$('form').submit(**function**(){

**var** msg = $('span').text();

**if**(msg != '사용가능한 사번입니다'){

alert('사용가능한 사번을 입력하세요');

frm.empno.focus();

**return** **false**;

}

});

});

**function** chk(){

**if**(!frm.empno.value){

alert('사번을 입력 후 중복체크하세요');

frm.empno.focus();

}**else**{

//var mgr = frm.mgr.options[frm.mgr.selectedIndex].value;

location.href = 'confirmNo.do?empno='+frm.empno.value;

}

}

</script>

</head>

<body>

<c:if test=*"*${**not empty** writeResult }*"*>

<script>alert('${writeResult}');</script>

</c:if>

<form action=*"write.do"* method=*"post"* name=*"frm"*>

<table>

<caption>사원등록</caption>

<tr>

<th>사번</th>

<td>

<input type=*"number"* name=*"empno"* required=*"required"* value=*"*${param.empno }*"*>

<input type=*"button"* value=*"중복확인"* onclick="chk()" class=*"btn"*><br>

<span><c:if test=*"*${msg!=null }*"*>${msg }</c:if></span>

</td>

</tr>

<tr><th>이름</th><td><input type=*"text"* name=*"ename"*></td></tr>

<tr><th>업무</th><td><input type=*"text"* name=*"job"*></td></tr>

<tr>

<th>관리자</th>

<td>

<select name=*"mgr"*>

<c:forEach var=*"mgrEmp"* items=*"*${managerList}*"*>

<option value=*"*${mgrEmp.empno }*"*>${mgrEmp.empno } - ${mgrEmp.ename }</option>

</c:forEach>

</select>

</td>

</tr>

<tr><th>입사일</th><td><input type=*"date"* name=*"hiredate"* required=*"required"* value=*"*<%=**new** Date(System.currentTimeMillis())%>*"*></td></tr>

<tr><th>급여</th><td><input type=*"number"* name=*"sal"* value=*"800"*></td></tr>

<tr><th>상여</th><td><input type=*"number"* name=*"comm"* value=*"0"*></td></tr>

<tr>

<th>부서코드</th>

<td>

<select name=*"deptno"*>

<c:forEach var=*"dept"* items=*"*${deptList }*"*>

<option value=*"*${dept.deptno }*"*>${dept.deptno }-${dept.dname }-${dept.loc }</option>

</c:forEach>

</select>

</td>

</tr>

<tr><td colspan=*"2"*><input type=*"submit"* value=*"등록"* class=*"btn"*></td></tr>

</table>

</form>

</body>

</html>

**update.jsp**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<%@ taglib prefix=*"fmt"* uri=*"http://java.sun.com/jsp/jstl/fmt"*%>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*${conPath }*/css/style.css"* rel=*"stylesheet"*>

<script src=*"https://code.jquery.com/jquery-3.6.0.js"*></script>

<link rel=*"stylesheet"* href=*"//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css"*>

<script src=*"https://code.jquery.com/jquery-1.12.4.js"*></script>

<script src=*"https://code.jquery.com/ui/1.12.1/jquery-ui.js"*></script>

<script>

$( **function**() {

$( "#datepicker" ).datepicker(

{ dateFormat : 'yy-mm-dd',

changeMonth: **true**, // 월을 바꿀수 있는 셀렉트 박스를 표시한다.

changeYear: **true**, // 년도를 바꿀 수 있는 셀렉트 박스를 표시한다.

monthNames : ['1월','2월','3월','4월','5월','6월','7월','8월','9월','10월','11월','12월'],

showMonthAfterYear : **true**,

yearSuffix : '년',

showOtherMonths : **true**, // 다른 달도 화면에 출력

dayNamesMin : ['일','월','화','수','목','금','토']

}

);

} );

</script>

</head>

<body>

<c:if test=*"*${**not empty** updateResult }*"*>

<script>alert('${updateResult}');</script>

</c:if>

<form action=*"update.do"* method=*"post"*>

<input type=*"hidden"* name=*"pageNum"* value=*"*${param.pageNum }*"*>

<input type=*"hidden"* name=*"empno"* value=*"*${empDto.empno }*"*>

<table>

<caption>${empDto.empno }번 사원 정보 수정</caption>

<tr>

<th>이름</th>

<td><input type=*"text"* name=*"ename"* value=*"*${empDto.ename }*"* required=*"required"*></td>

</tr>

<tr>

<th>업무</th>

<td><input type=*"text"* name=*"job"* value=*"*${empDto.job }*"* required=*"required"*></td>

</tr>

<tr>

<th>관리자사번</th>

<td><input type=*"number"* name=*"mgr"* value=*"*${empDto.mgr }*"* required=*"required"*></td>

</tr>

<tr>

<th>입사일</th>

<td><input type=*"text"* name=*"hiredate"* value=*"*${empDto.hiredate.toString().substring(0,10) }*"* required=*"required"* id=*"datepicker"*></td>

</tr>

<tr>

<th>급여</th>

<td><input type=*"number"* name=*"sal"* value=*"*${empDto.sal }*"* required=*"required"*></td>

</tr>

<tr>

<th>상여</th>

<td><input type=*"number"* name=*"comm"* value=*"*${empDto.comm }*"* required=*"required"*></td>

</tr>

<tr>

<th>부서코드</th>

<td><input type=*"number"* name=*"deptno"* value=*"*${empDto.deptno }*"* required=*"required"*></td>

</tr>

<tr>

<td colspan=*"2"*>

<input type=*"submit"* value=*"수정"* class=*"btn"*>

<input type=*"button"* value=*"목록"* class=*"btn"* onclick=*"location.href='empDeptList.do?pageNum=*${param.pageNum}*'"*>

</td>

</tr>

</table>

</form>

</body>

</html>

XML에서 동적 SQL 처리

OGNL(Object Graph Navigation Language)

(ex) <if ~ <forEach~

* if 엘리먼트 : 모든 조건에 대해 처리하고 만족하는 조건의 결과를 모두 적용한다.
* 동적 QUERY : 위와 같이 SQL 구문이 조건에 따라서 변경되는 것을 말한다.

<select id=”selData”>

SELECT \* FROM EMP

<if test=”ename != null”> WHERE ENAME=#{ENAME} </if>

</select>

* 여러 논리 조건

<if test=”ename!=null and sal !=0”>

* EMP 리스트 데이터 중, ename이나 job 등 해당 데이터가 있을 때, 검색이 되고 그렇지 않으면 전체 검색이 되는 동적 query 문을 작성하세요(조건으로 ename만 있을 때, job만 있을 때, ename과 job이 있을 때, 전체 조건이 없을 때 모두 생각하기)

<select id=*"empList"* resultMap=*"EmpMap"* parameterType=*"Emp"*>

select \* from emp where 1=1

<if test=*"ename!=null and ename!='' "*>

and ename like '%'||#{ename}||'%'

</if>

<if test=*"job!=null and job!='' "*>

and ename like '%'||#{ename}||'%'

</if>

</select>

* choose(when, otherwise) : 조건 지정과 그 외에 대한 처리를 동적인 SQL에서 진행할 때 활용된다

<select id=*"empList"* resultMap=*"EmpMap"* parameterType=*"Emp"*>

select \* from emp where 1=1

<choose>

<when test=*"sal>5000"*> and deptno=10 </when>

<when test=*"sal>4000"*> and deptno=20 </when>

<when test=*"sal>1000"*> and deptno=30 </when>

</choose>

</select>

* trim(where) 엘리먼트 : if 엘리먼트가 조건 처리에서 where 구문에 대한 처리를 보완하기 위해 사용한다.
  + Prefix : 처리 후 엘리먼트 내용이 있으면 앞에 붙여줌
  + prefixOverrides : 처리 후 앞에 해당 문자 있으면 자동으로 지워줌

Select \* from emp

// Where 1=1 구분의 활용하지 않으면

<where>

<trim prefix=”WHERE” prefixOverrides=”AND | OR”>

<if test=”ename != null>ename=#{ename}</if>

* foreach 엘리먼트 : 동적 SQL을 loop문으로 반복처리 해야 할 경우
  + 조회조건에서 부서를 여러 개 검색 조건으로 처리할 때..

SELECT \* FROM EMP WHERE DEPTNO IN(10,20,30);

* + 속성 <foreach collection=”” item=””

Collection : 배열의 목록 리스트를 가져옮 ex) jstl items

Item : 배열의 목록 단위 (ex) jstl의 var

Index : 몇 번째 값인지 나타내는 인덱스. 0부터 시작

Select \* from emp where 1=1

<if test=”deptnos != null”> -> getDeptnos()가 있어야 함

Deptno in <foreach collection=”deptnos” item=”chdeptno”