

[회사 이름 입력]

MyBatis 기본설정 및 예제

[문서 부제 입력]

CHO HANJOONG

2012-10-14

1. 문서이력	2
2. Mybatis3.1과 JSP3.0 연동하기	3
2-1. mybatis참고 URL 및 라이브러리 다운로드	3
2-2. 예제 소스 다운로드	3
2-3. 예제소스 구조	4
2-4. 예제소스 설명	4
2-4-1. 개요	4
2-4-2. 예제 실행 화면	5
2-4-3. 주요설정 파일 및 코드 설명	8
2-4-3-1. db.properties	8
2-4-3-2. SqlSessionFactoryManager.java	9
2-4-3-3. SqlMapConfig.xml	10
2-4-3-4. Day1.xml	11
2-4-3-5. Day2.xml	12
2-4-3-6. Day1Dao	13
2-4-3-7. Day2Dao	14
2-4-3-8. day1.jsp	15
2-4-3-9. Day2.jsp	16
3. Mybatis3.1과 Spring3.1 연동하기	18
4. MyBatis-Spring1.1과 Spring3.1 연동하기	18

1. 문서이력

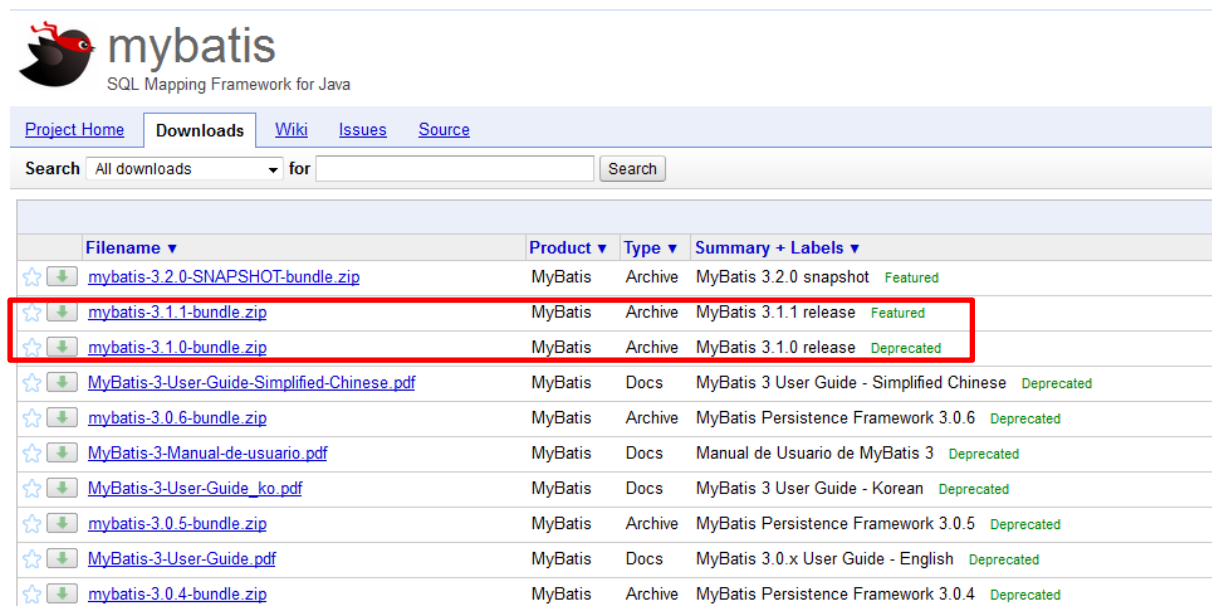
No	Version	작성자	작성일자	변경사유	변경내용
1	1.0	조 한 중	2012년10월14일	최초작성	
2					
3					
4					
5					
6					
7					

2. Mybatis3.1과 JSP3.0 연동하기

2-1. mybatis참고 URL 및 라이브러리 다운로드

아래의 url에서 원하는 버전의 라이브러리를 다운받아 압축을 해제한다.

<http://code.google.com/p/mybatis/downloads/list?can=1>



Filename ▼	Product ▼	Type ▼	Summary + Labels ▼
mybatis-3.2.0-SNAPSHOT-bundle.zip	MyBatis	Archive	MyBatis 3.2.0 snapshot Featured
mybatis-3.1.1-bundle.zip	MyBatis	Archive	MyBatis 3.1.1 release Featured
mybatis-3.1.0-bundle.zip	MyBatis	Archive	MyBatis 3.1.0 release Deprecated
MyBatis-3-User-Guide-Simplified-Chinese.pdf	MyBatis	Docs	MyBatis 3 User Guide - Simplified Chinese Deprecated
mybatis-3.0.6-bundle.zip	MyBatis	Archive	MyBatis Persistence Framework 3.0.6 Deprecated
MyBatis-3-Manual-de-usuario.pdf	MyBatis	Docs	Manual de Usuario de MyBatis 3 Deprecated
MyBatis-3-User-Guide_ko.pdf	MyBatis	Docs	MyBatis 3 User Guide - Korean Deprecated
mybatis-3.0.5-bundle.zip	MyBatis	Archive	MyBatis Persistence Framework 3.0.5 Deprecated
MyBatis-3-User-Guide.pdf	MyBatis	Docs	MyBatis 3.0.x User Guide - English Deprecated
mybatis-3.0.4-bundle.zip	MyBatis	Archive	MyBatis Persistence Framework 3.0.4 Deprecated

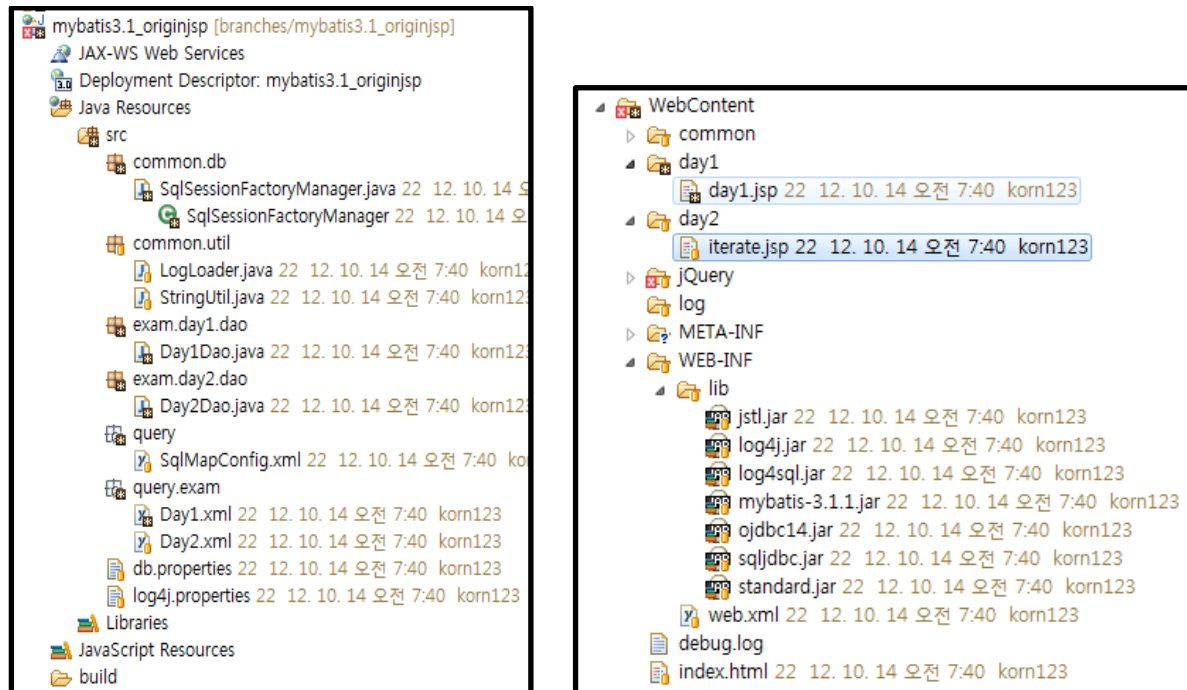
2-2. 예제 소스 다운로드

블로그 첨부파일 -

svn repository – https://dev.naver.com/svn/samplesource/branches/mybatis3.1_originjsp

2-3. 예제소스 구조

이번 예제는 mybatis3.1과 jsp3.0을 연동하는 예제로 웹프레임워크를 이용하지 않는 간편한 예제소스파일의 구조는 아래와 같다.



2-4. 예제소스 설명

2-4-1. 개요

기본컨트롤 및 뷰 구현은 JSP에서 모두 처리되며 데이터베이스와 연동하는 모델은 java코드에서 처리한다.

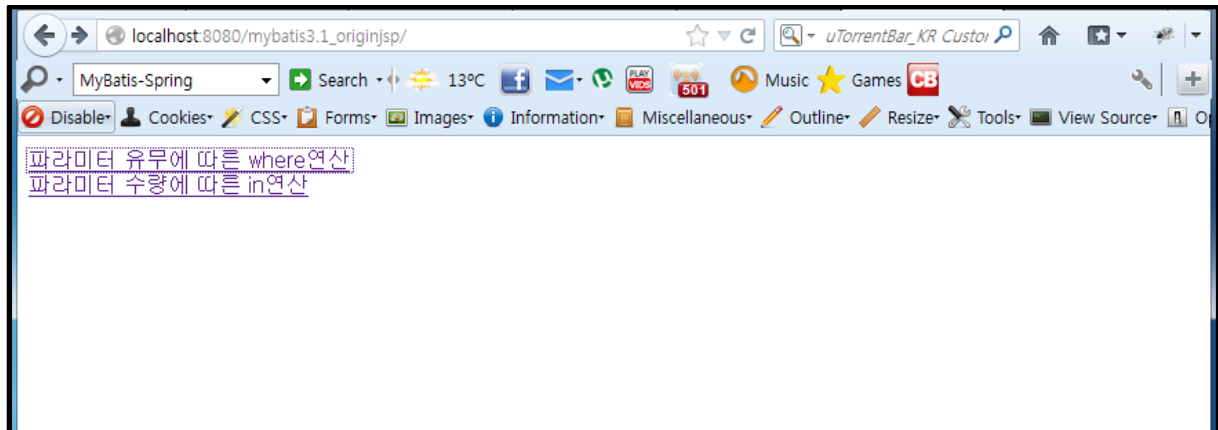
SessionFactoryManager.java클래스에서 SqlSessionFactory인스턴스를 생성하며 이렇게 생성된 인스턴스를 Day1Dao.java / Day2Dao.java에서 취득하여 쿼리를 수행한다.

예제코드는 파라미터 유무에 따른 where연산과 파라미터 수량에 따른 in연산을 수행하는 두 가지 형태의 코드로 구성되어 있다.

Database는 Oracle XE에 기본적으로 생성되어 있는 HR스키마를 이용한다.

2-4-2. 예제 실행 화면

<메인화면>

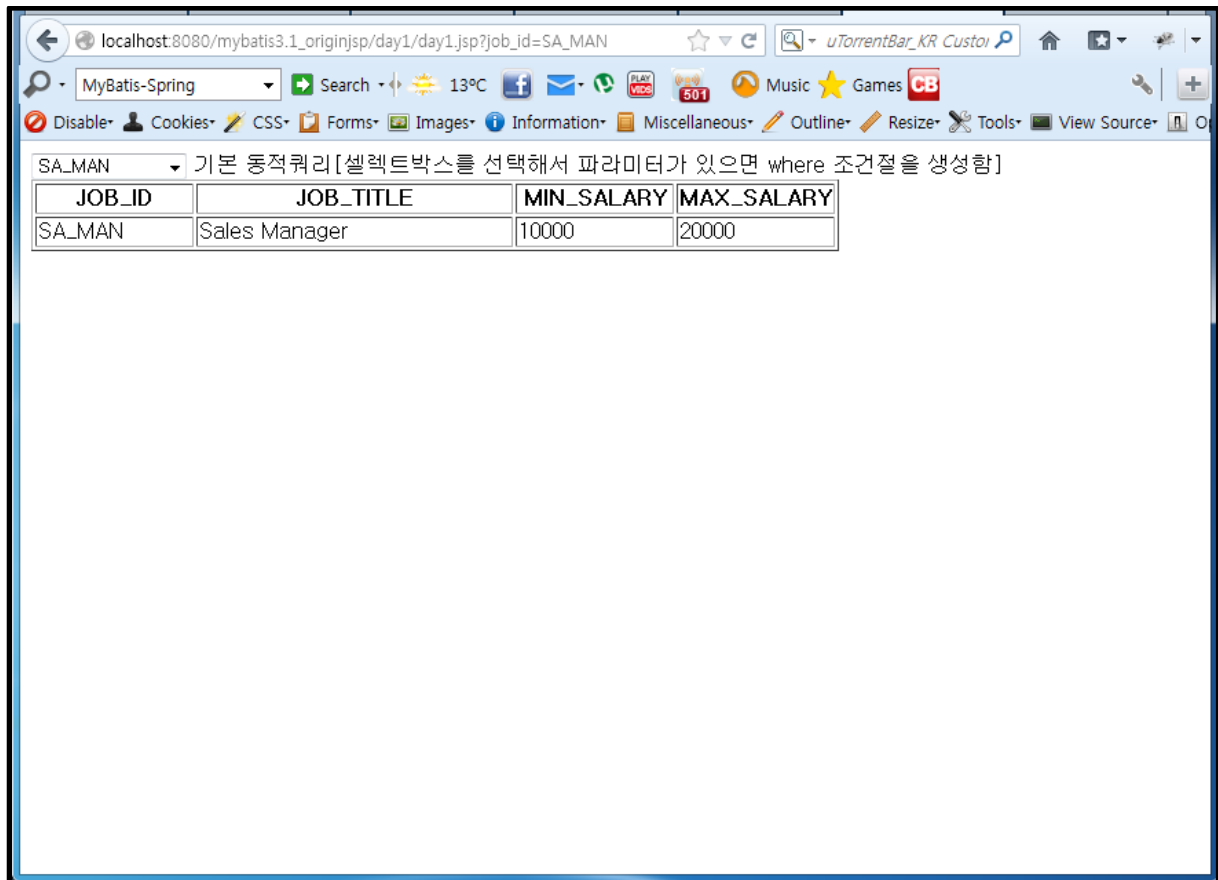


<where조건절 없이 디폴트값으로 모두 선택된 화면>



http://blog.naver.com/korn123

<콤보박스 선택값으로 where조건절 생성화면>



http://blog.naver.com/korn123

<체크박스를 선택하기 전 기본화면>

localhost:8080/mybatis3.1_originjsp/day2/iterate.jsp

MyBatis-Spring

13°C

501

Music Games CB

Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
--------	-----------	------------	------------

- ☐ President
- ☐ Administration Vice President
- ☐ Administration Assistant
- ☐ Finance Manager
- ☐ Accountant
- ☐ Accounting Manager
- ☐ Public Accountant
- ☐ Sales Manager
- ☐ Sales Representative
- ☐ Purchasing Manager
- ☐ Purchasing Clerk
- ☐ Stock Manager
- ☐ Stock Clerk
- ☐ Shipping Clerk
- ☐ Programmer
- ☐ Marketing Manager
- ☐ Marketing Representative
- ☐ Human Resources Representative
- ☐ Public Relations Representative

검색

http://blog.naver.com/korn123

<체크박스 선택 후 해당 파라미터 집합으로 in연산을 수행한 화면>

The screenshot shows a web browser window with the URL `localhost:8080/mybatis3.1_originjsp/day2/iterate.jsp?jobId=AD_PRES&jobId=...`. The page displays a table with job information and a list of checkboxes for selection.

JOB_ID	JOB_TITLE	MIN_SALARY	MAX_SALARY
AD_ASST	Administration Assistant	3000	6000
AD_PRES	President	20000	40000
AD_VP	Administration Vice President	15000	30000

Below the table, there is a list of checkboxes for selection:

- ☒ President
- ☒ Administration Vice President
- ☒ Administration Assistant
- ☐ Finance Manager
- ☐ Accountant
- ☐ Accounting Manager
- ☐ Public Accountant
- ☐ Sales Manager
- ☐ Sales Representative
- ☐ Purchasing Manager
- ☐ Purchasing Clerk
- ☐ Stock Manager
- ☐ Stock Clerk
- ☐ Shipping Clerk
- ☐ Programmer
- ☐ Marketing Manager
- ☐ Marketing Representative
- ☐ Human Resources Representative
- ☐ Public Relations Representative

A "검색" (Search) button is located at the bottom of the list.

2-4-3. 주요설정 파일 및 코드 설명

2-4-3-1. db.properties

SqlMapConfig.xml에서 datasource설정시 사용되는 DB Connection정보를 관리한다.

```
#Oracle
#driver = oracle.jdbc.driver.OracleDriver
driver = core.log.jdbc.driver.OracleDriver
url    = jdbc:oracle:thin:@127.0.0.1:1521:XE
user   = hr
pwd    = hr
```

2-4-3-2. SqlSessionFactoryManager.java

SqlMapConfig.xml 파일을 로딩하여 SqlSessionFactory 인스턴스를 싱글톤으로 생성한다.

```
/**
 * SqlSessionFactory 인스턴스를 싱글톤으로 반환함
 * @author hanjoong
 */
public class SqlSessionFactoryManager {

    private static Logger Log =
        Logger.getLogger(SqlSessionFactoryManager.class);
    private static final SqlSessionFactory sqlMapper;

    static{
        String resource = "query/SqlMapConfig.xml";
        Reader reader = null;
        try {
            reader = Resources.getResourceAsReader(resource);
        } catch (IOException e) {
            Log.debug("초기화 에러");
        }
        sqlMapper = new SqlSessionFactoryBuilder().build(reader);
    }

    /**
     *
     * @return SqlSessionFactory 인스턴스를 반환
     */
    public static SqlSessionFactory getSqlSessionFactory() {
        return sqlMapper;
    }
}
```

2-4-3-3. SqlMapConfig.xml

Mybatis 사용을 위한 설정관리 파일로 datasource정보 및 mapper파일 경로를 지정한다.

```
<?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>
  <properties resource="db.properties">
  </properties>
  <environments default="development" >
    <environment id="development" >
      <transactionManager type="JDBC" />
      <dataSource type="POOLED" >
        <property name="driver" value="${driver}" />
        <property name="url" value="${url}" />
        <property name="username" value="${user}" />
        <property name="password" value="${pwd}" />
      </dataSource>
    </environment>
  </environments>
  <mappers>
    <mapper resource="query/exam/Day1.xml" />
    <mapper resource="query/exam/Day2.xml" />
  </mappers>
</configuration>
```

2-4-3-4. Day1.xml

실질적인 쿼리를 수행하기 위한 Mapper설정 파일

```
<?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="query.exam.Day1">
    <resultMap id="hashmap" type="java.util.HashMap">
    </resultMap>

    <!-- JOBS 테이블의 모든 로우정보를 받아옴 -->
    <select id="getJobs" parameterType="int" resultMap="hashmap">
        SELECT * FROM JOBS
        <where>
            <if test="job_id != null">
                JOB_ID = #{job_id}
            </if>
        </where>
    </select>

</mapper>
```

2-4-3-5. Day2.xml

실질적인 쿼리를 수행하기 위한 Mapper설정 파일

```
<?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="query.exam.Day2">
    <resultMap id="hashmap" type="java.util.HashMap">
    </resultMap>

    <!-- JOBS 테이블의 모든 로우정보를 받아옴 -->
    <select id="getJobs" parameterType="int" resultMap="hashmap">
        SELECT * FROM JOBS
        <where>
            <if test="job_id != null">
                JOB_ID = #{job_id}
            </if>
        </where>
    </select>

    <!-- Check된 JOB_ID와 일치하는 정보를 반환 -->
    <select id="getCheckJobs" resultMap="hashmap"
        parameterType="java.util.HashMap">
        SELECT * FROM JOBS
        <where>
            <if test="jobIdArr != null">
                <foreach collection="jobIdArr" open="JOB_ID IN (" close=")"
                    separator="," item="jobIdArr">
                        #{jobIdArr}
                    </foreach>
            </if>
        </where>
    </select>
</mapper>
```

2-4-3-6. Day1Dao

JSP에서 넘겨주는 정보를 mapper와 연결한다.

```
public class Day1Dao {

    SqlSessionFactory sqlMapper =
    SqlSessionFactoryManager.getSqlSessionFactory();

    public Day1Dao() {
        // TODO Auto-generated constructor stub
    }

    /**
     * JOBS 테이블의 정보를 반환(검색키워드가 없으면 모두 반환)
     * @param map
     * @return
     */
    @SuppressWarnings({ "rawtypes", "unchecked" })
    public ArrayList<HashMap<String, String>>
        getJobs(HashMap<String, Object> map){
        ArrayList<HashMap<String, String>> list = null;
        list = (ArrayList)sqlMapper.openSession().
            selectList("query.exam.Day1.getJobs",map);

        return list;
    }

    /**
     * JOB_ID를 모두 반환함
     * @return
     */
    @SuppressWarnings({ "rawtypes", "unchecked" })
    public ArrayList<HashMap<String, String>> getJobId(){
        ArrayList<HashMap<String, String>> list = null;
        HashMap<String, Object> map = null;
        list = (ArrayList)sqlMapper.openSession().
            selectList("query.exam.Day1.getJobs",map);

        return list;
    }
}
```

2-4-3-7. Day2Dao

JSP에서 넘겨주는 정보를 mapper와 연결한다.

```
public class Day2Dao {

    SqlSessionFactory sqlMapper =
    SqlSessionFactoryManager.getSqlSessionFactory();

    public Day2Dao() {
        // TODO Auto-generated constructor stub
    }

    /**
     * JOBS 테이블의 정보를 반환(String[] 형태의 키값을 받아서 필터링함)
     * @param map
     * @return
     */
    @SuppressWarnings({ "rawtypes", "unchecked" })
    public ArrayList<HashMap<String, String>> getJobs(String[] jobIdArr){
        ArrayList<HashMap<String, String>> list = null;
        HashMap<String, Object> map = new HashMap<String, Object>();
        map.put("jobIdArr",jobIdArr);
        list = (ArrayList)sqlMapper.openSession().
            selectList("query.exam.Day2.getCheckJobs", map);
        return list;
    }

    /**
     * JOB_ID를 모두 반환함
     * @return
     */
    @SuppressWarnings({ "rawtypes", "unchecked" })
    public ArrayList<HashMap<String, String>> getJobId(){
        ArrayList<HashMap<String, String>> list = null;
        HashMap<String, Object> map = null;
        list = (ArrayList)sqlMapper.openSession().
            selectList("query.exam.Day2.getJobs",map);
        return list;
    }
}
```

2-4-3-8. day1.jsp

사용자의 요청을 service로직에 전달하고 jstl 및 el을 이용하여 반환된 값을 보여준다.

```
<%@page import="common.util.StringUtil"%>
<%@ page import="java.util.HashMap"%>
<%@ page import="java.util.ArrayList"%>
<%@ page import="exam.day1.dao.Day1Dao"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%
    String job_id = StringUtil.nullToEmpty(request.getParameter("job_id"));
    Day1Dao dao = new Day1Dao();
    ArrayList<HashMap<String, String>> list = null;
    ArrayList<HashMap<String, String>> jobIdList = dao.getJobId(); //선택옵션 생성
    if("").equals(job_id)){ //검색 키워드가 없을경우
        list = dao.getJobs(null);
    }else{
        //검색 키워드가 있을경우
        HashMap<String, Object> map = new HashMap<String, Object>();
        map.put("job_id", job_id);
        list = dao.getJobs(map);
    }
%>
<c:set var="job_id" value="<%=job_id%>"></c:set>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>DAY - 1</title>
<script type="text/javascript">
function searchGo() {
    var job_id = document.getElementById('job_id').value;
    location.href = 'day1.jsp?job_id=' + job_id;
}
</script>
</head>
<body>

<div>
<select onchange="searchGo();" id="job_id" name="job_id">
    <option value="">전체조회</option>
    <c:forEach var="jobIdList" items="<%=jobIdList%>">
        <option value="{jobIdList.JOB_ID}" <c:if test = "{jobIdList.JOB_ID ==
            job_id}"> selected="selected" </c:if> >{jobIdList.JOB_ID}
        </option>
    </c:forEach>
</select>
    기본 동적쿼리[셀렉트박스를 선택해서 파라미터가 있으면 where 조건절을 생성함]
<table border="1">
```



```
<colgroup>
  <col width="20%">
  <col width="40%">
  <col width="20%">
  <col width="20%">
</colgroup>

<thead>
  <tr>
    <th>JOB_ID</th><th>JOB_TITLE</th><th>MIN_SALARY</th><th>MAX_SALARY</th>
  </tr>
</thead>

<tbody>
  <!-- jstl을 이용하여 list에 담겨있는 Map의 value값을 호출 -->
  <c:forEach var="list" items="<%=list%">
    <tr>
      <td>${list.JOB_ID}</td>
      <td>${list.JOB_TITLE}</td>
      <td>${list.MIN_SALARY}</td>
      <td>${list.MAX_SALARY}</td>
    </tr>
  </c:forEach>
</tbody>
</table>
</div>

</body>
</html>
```

2-4-3-9. Day2.jsp

사용자의 요청을 service로직에 전달하고 jstl 및 el을 이용하여 반환된 값을 보여준다.

```
<%@ page import="java.util.HashMap"%>
<%@ page import="java.util.ArrayList"%>
<%@ page import="exam.day2.dao.Day2Dao"%>
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<%
    Day2Dao dao = new Day2Dao();
    //선택옵션 생성
    ArrayList<HashMap<String, String>> jobIdList = dao.getJobId();
    ArrayList<HashMap<String, String>> list =
        new ArrayList<HashMap<String, String>>();
    String[] checkIds = request.getParameterValues("jobId");
    if(checkIds != null){ //체크박스가 하나 이상 선택될경우
        list = dao.getJobs(checkIds);
    }
}
```

```
%>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>동적쿼리 iterate</title>
</head>
<body>

<!-- 체크 영역 -->
<div style="float: left;">
  <table border="1">
    <colgroup>
      <col width="20%">
      <col width="40%">
      <col width="20%">
      <col width="20%">
    </colgroup>

    <thead>
      <tr>
        <th>JOB_ID</th><th>JOB_TITLE</th><th>MIN_SALARY</th><th>MAX_SALARY</th>
      </tr>
    </thead>

    <tbody>
      <!-- jstl을 이용하여 list에 담겨있는 Map의 value값을 호출 -->
      <c:forEach var="list" items="%=list%">
        <tr>
          <td>${list.JOB_ID}</td>
          <td>${list.JOB_TITLE}</td>
          <td>${list.MIN_SALARY}</td>
          <td>${list.MAX_SALARY}</td>
        </tr>

      </c:forEach>
    </tbody>
  </table>

</div>

<!-- 선택 영역 -->
<div style="float: right;">
  <form action="" method="get">
    <c:forEach var="jobIdList" items="%=jobIdList%" >
      <input type="checkbox" name="jobId" value="${jobIdList.JOB_ID}"
        />${jobIdList.JOB_TITLE} <br />
    </c:forEach>

    <input type="submit" value="검색" />
  </form>
</div>
</body>
</html>
```

<http://blog.naver.com/korn123>

```
</div>
```

```
</body>
```

```
</html>
```

3. Mybatis3.1과 Spring3.1 연동하기

4. MyBatis-Spring1.1과 Spring3.1 연동하기