

Ibatis/Mybatis

강사 : 강병준

ORM 프레임워크인 iBATIS

- ORM(Object Relational Mapping) 프레임워크는 데이터베이스와 객체와의 관계를 맵핑시켜 퍼시스턴스 로직처리를 도와주는 프레임워크이다.
- XML 파일에 맵핑 정보를 기술하여 데이터베이스의 테이블과 자바 객체를 맵핑하여 데이터베이스에 생성, 조회, 수정, 삭제(CRUD)작업을 도와주는 역할을 한다.
- ORM 프레임워크는 퍼시스턴스 층에 생산성을 높여 주기 위한 프레임워크로 JDBC를 사용할 때보다 약 60% 정도의 코드만으로 프로그램 작성이 가능해 진다.

작업 절차

- JDBC 드라이버 설치
- DBCP 사용을 위한 jar 파일 설치하기
- iBATIS 설치하기
- iBATIS 사용한 JDBC 프로젝트 작성하기
- 데이터베이스 연결 정보를 담은 프로퍼티 파일 작성하기
- SQLMaps 설정파일인 SqlMapConfig.xml 파일 작성하기

iBATIS 다운받아 설치하기

- <http://ibatis.apache.org/>
- 2010년에 mybatis로 변경



MyBatis

- ❖ 마이바티스는 개발자가 지정한 SQL, 저장프로시저를 지원하는 퍼시스턴스 프레임워크
- ❖ 마이바티스는 JDBC로 처리하는 상당부분의 코드와 파라미터 설정 및 결과 매핑을 대신 수행해줍니다.
- ❖ 마이바티스는 데이터베이스 레코드에 원시타입과 Map 인터페이스 그리고 자바 POJO(Plain Old Java Object) 를 설정해서 매핑하기 위해 XML과 애노테이션을 사용할 수 있습니다.

- myBatis 다운로드: <http://blog.mybatis.org/p/products.html>

- Maven Dependency

```
<dependency>  
    <groupId>org.mybatis</groupId>  
    <artifactId>mybatis</artifactId>  
    <version>3.3.1</version>  
</dependency>
```

MyBatis

※ MyBatis의 특징

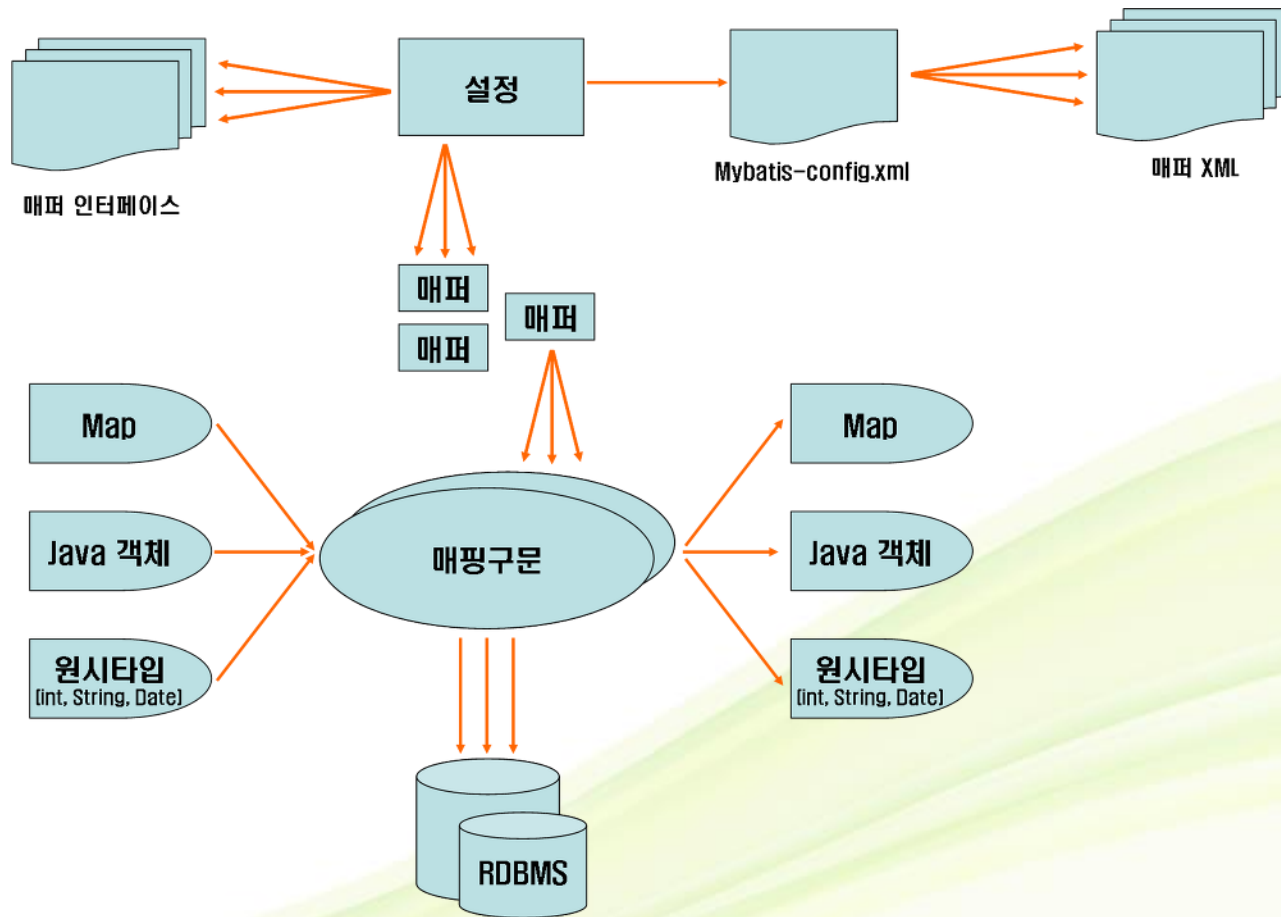
- ❖ JDBC 프레임워크
- ❖ SQL은 개발자가 작성하고 MyBatis는 JDBC를 이용해서 이를 실행하는 역할
- ❖ ORM(Object Relation Mapping)이 아닌 SQL 매퍼
- ❖ ORM은 데이터베이스 객체를 자바 객체로 매핑해서 객체 간의 관계를 바탕으로 SQL을 자동으로 생성해주는데 대표적인 프레임워크가 하이버네이트
- ❖ SQL 문장과 파라미터를 PreparedStatement를 이용해서 자동 매핑
- ❖ Select 구문의 결과를 객체(DTO 또는 Map)로 자동 매핑
- ❖ 트랜잭션 관리가 용이
- ❖ 스프링과 같은 외부 트랜잭션 관리자를 사용할 수 있도록 한다.
- ❖ 스프링 연동 모듈을 제공해서 스프링과 연동 가능
- ❖ JVM위에서 동작하는 함수형 언어인 스칼라와 마이크로소프트 사의 닷넷 그리고 루비를 지원

MyBatis

❖ 마이바티스 구조

- 설정 파일: 데이터베이스 설정과 트랜잭션 등 마이바티스가 동작하는 규칙을 정의
- 매퍼: SQL을 XML에 정의한 매퍼 XML 파일 또는 SQL을 인터페이스 메소드에 annotation으로 정의한 매퍼 인터페이스
- 결과 매핑과 매핑 구문: 조회 결과를 자바 객체에 설정하는 규칙을 나타내는 결과 매핑과 SQL을 XML에 정의한 매핑 구문
- 지원하는 파라미터 타입: Map 객체, 자바 클래스, 원시 타입(Wrapper Class, String)
- 지원하는 결과 타입: 파라미터 타입과 동일

MyBatis



MyBatis 환경 설정

```
<?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>
  <environments default="development">
    <environment id="development">
      <!-- ① 트랜잭션 관리자 -->
      <transactionManager type="JDBC" />
      <!-- ② 데이터베이스 설정 -->
      <dataSource type="POOLED">
        <property name="driver" value="oracle.jdbc.driver.OracleDriver" />
        <property name="url" value="jdbc:oracle:thin:@localhost:1521:mybatis" />
        <property name="username" value="mybatis" />
        <property name="password" value="asdf" />
      </dataSource>
    </environment>
  </environments>
  <!-- ③ 매퍼정보 설정 -->
  <mappers>
    <mapper resource="ldg/mybatis/repository/mapper/CommentMapper.xml" />
  </mappers>
</configuration>
```


MyBatis 환경 설정

① 트랜잭션 관리자

마이바티스는 두가지 타입의 TransactionManager를 제공

- JDBC - 이 설정은 간단하게 JDBC 커밋과 롤백을 처리하기 위해 사용

- MANAGED

- ❖ 이 설정은 어떤 것도 하지 않습니다.
- ❖ 결코 커밋이나 롤백을 하지 않습니다.
- ❖ 대신 컨테이너가 트랜잭션의 모든 생명주기를 관리합니다.

- 마이바티스를 스프링과 함께 사용하는 경우에는 구성할 필요가 없습니다 스프링 모듈 자체의 설정 때문에 어떤 TransactionManager 이전에 설정된 구성을 무시합니다.

MyBatis 환경 설정

② 데이터베이스 설정

dataSource엘리먼트는 표준 JDBC DataSource인터페이스를 사용하여 JDBC Connection객체의 소스를 설정합니다.

❖ UNPOOLED - 이 구현체는 매번 요청에 대해 커넥션을 열고 닫는 간단한 DataSource로 느리기는 하지만 성능을 크게 필요로 하지 않는 간단한 애플리케이션에 적합하며 5개의 프로퍼티로 설정한다.

- driver - JDBC드라이버의 패키지 경로를 포함한 결제 자바 클래스명
- url - 데이터베이스 인스턴스에 대한 JDBC URL
- username - 데이터베이스에 로그인 할 때 사용할 사용자명
- password - 데이터베이스에 로그인 할 때 사용할 패스워드

MyBatis 환경 설정

- ❖ POOLED - DataSource에 풀링이 적용된 JDBC 커넥션을 위한 구현체로 새로운 Connection 인스턴스를 생성하기 위해 매번 초기화하는 것을 피하게 해주는 방식으로 빠른 응답을 요구하는 웹 애플리케이션에서 사용
 - poolMaximumActiveConnections - 주어진 시간에 존재할 수 있는 활성화된(사용중인) 커넥션의 수. 디폴트는 10
 - poolMaximumIdleConnections - 주어진 시간에 존재할 수 있는 유휴 커넥션의 수
 - poolTimeToWait - 풀이 로그 상태를 출력하고 비정상적으로 긴 경우 커넥션을 다시 얻으려고 시도하는 로우 레벨 설정. 디폴트는 20000ms(20 초)
 - poolPingEnabled - 핑쿼리를 사용할지 말지를 결정. 사용한다면 오류가 없는(그리고 빠른) SQL 을 사용하여 poolPingQuery 프로퍼티를 설정해야 합니다. 디폴트는 false
 - poolPingConnectionsNotUsedFor - poolPingQuery가 얼마나 자주 사용될지 설정하는 것으로 디폴트는 0

MyBatis 환경 설정

- 데이터베이스 접속 드라이버와 URL작성 방법

Oracle

드라이버 클래스명: oracle.jdbc.driver.OracleDriver

JDBC URL: jdbc:oracle:thin:@localhost:1521:SID이름

MySQL

드라이버 클래스명: com.mysql.jdbc.Driver

JDBC URL: jdbc:mysql://localhost:3306/데이터베이스이름

MSSQL

드라이버 클래스명: com.microsoft.jdbc.sqlserver.SQLServerDriver

JDBC URL: jdbc:microsoft:sqlserver://203.247.166.172;databasename=데이터베이스이름

MyBatis 환경 설정

③ 매퍼정보 설정

- SQL을 선언해둔 XML 파일의 위치나 인터페이스 형태의 매퍼 위치를 지정해줘야 합니다.
- XML 위치는 클래스 패스를 기준으로 지정하면 됩니다.

```
<mappers>
```

```
  <mapper resource="org/mybatis/builder/AuthorMapper.xml"/>
```

```
  <mapper resource="org/mybatis/builder/BlogMapper.xml"/>
```

```
  <mapper resource="org/mybatis/builder/PostMapper.xml"/>
```

```
</mappers>
```

```
<mappers>
```

```
  <mapper class="org.mybatis.builder.AuthorMapper"/>
```

```
  <mapper class="org.mybatis.builder.BlogMapper"/>
```

```
  <mapper class="org.mybatis.builder.PostMapper"/>
```

```
</mappers>
```

MyBatis 환경 설정

※ MyBatis의 매퍼 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="매퍼이름">
  <resultMap id="별명" type=" 실제 자료형">
    </resultMap>

  <select id=" 호출할 이름 " resultMap=" 결과 자료형" 또는 resultType="자료형" >
    select 구문
  </select>
  <insert id=" 호출할 이름" parameterType=" 매개변수 자료형">
    INSERT INTO
      AddressBook
      VALUES ({프로퍼티이름 또는 키이름})
  </insert>

</mapper>
```

MyBatis 환경 설정

※ MyBatis의 자료형 매핑

Number – Integer, Double, Float, Byte, Short, Long, BigDecimal

Char, Varchar, Clob – String

Date – java.sql.Date, java.util.Date

Time – java.sql.Time, java.util.Date

TimeStamp – java.sql.Time, stamp

Blob – byte []

MyBatis 환경 설정

※ MyBatis의 매퍼 파일 작성 시 유의 사항

- wildcard 문자를 이용해서 검색하는 경우에 필드명이 "title" 이고 해당 필드에서 검색할 내용을 "keyword"를 포함하는 내용이라고 하면 아래와 같이 검색할 수 있습니다.

[MySQL]

title like CONCAT('%',#{keyword},'%')

[Oracle]

title like '%' || #{keyword} || '%'

[MSSQL]

title like '%' + #{keyword} + '%'

- MyBatis에서 SQL을 작성할 때 비교 연산자를 직접 사용하면 에러

Where #{num} < 5

Mybatis mapper.xml안에서 비교 연산자(<= , >=, <, >) 을 써야 할 상황에서, 위와 같이 연산자를 직접 쓰게되면 오류 발생합니다. < 를 태그의 시작으로 인식해서 그렇습니다.

Where #{num} <![CDATA[<]> 5

MyBatis 환경 설정

- ❖ 모든 MyBatis 애플리케이션은 SqlSessionFactory 인스턴스를 사용합니다.
- ❖ SqlSessionFactory 인스턴스는 SqlSessionFactoryBuilder를 사용하여 만들 수 있습니다.
- ❖ SqlSessionFactoryBuilder는 XML설정파일에서 SqlSessionFactory 인스턴스를 빌드할 수 있습니다.

```
String resource = "환경설정파일 경로";  
Reader reader = Resources.getResourceAsReader(resource);  
SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder().build(reader);
```

위의 Factory를 이용해서 SqlSession 객체를 생성합니다.
위의 객체를 가지고 openSession()를 호출하면 SqlSession객체가 생성됩니다.

```
SqlSession sqlSession = sqlSessionFactory.openSession();
```

MyBatis 환경 설정

❖ sql 실행 메소드 호출

- ✓ sqlSession.insert, delete, update, selectOne, selectList(매퍼이름.id, 매개변수)를 호출하면 됩니다.
- ✓ 이 때 insert와 delete, update는 영향 받은 행의 개수를 정수로 리턴합니다.
- ✓ selectOne은 설정한 resultType으로 리턴
- ✓ selectList는 설정한 resultType의 List로 리턴
- ✓ selectOne 메소드는 결과가 없거나 2개 이상으로 나오면 예외를 발생
- ✓ Map으로 select 구문의 결과를 받는 경우 컬럼 이름이 키가 되고 데이터가 값
- ✓ 오라클의 데이터의 타입이 Number이면 Map에서는 BigDecimal로 저장
- ✓ Map으로 select 구문의 결과를 받는 경우 키의 이름은 오라클은 모두 대문자 MySQL은 소문자로 저장
- ✓ insert, delete, update 작업 시 autoCommit 되지 않으므로 commit()을 호출해서 직접 commit을 수행 해 주어야 합니다.

MyBatis 환경 설정

- ✓ sql 사용이 종료되면 close()를 호출해서 닫아 주어야 합니다.
- ✓ Java 1.7 & MyBatis 3.2 이상의 버전을 사용하면 아래처럼 작성해도 됩니다.

```
try (SqlSession session = sqlSessionFactory.openSession()) {  
    session.insert(...);  
    session.commit();  
}
```

Mybatis 구성

- myBatis
 - > Deployment Descriptor: myBatis
 - > JAX-WS Web Services
 - < Java Resources
 - < src
 - > myBatis.controller
 - > myBatis.dao
 - > myBatis.model
 - > myBatis.service
 - configuration.xml
 - db.properties
 - member.xml
 - > Libraries
 - > JavaScript Resources
 - > build
 - < WebContent
 - > META-INF
 - > WEB-INF
 - common.css
 - delete.jsp
 - index.jsp
 - join.jsp
 - joinForm.jsp

db.properties

driver=oracle.jdbc.OracleDriver
url=jdbc:oracle:thin:@localhost:1521:xe
username=scott
password=tiger

Mybatis

configuration.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>
  <properties resource="dbconnect.properties"/>
  <typeAliases>
    <typeAlias alias="member" type="ch17.model.Member" />
  </typeAliases>
  <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="${username}" />
        <property name="password" value="${password}" />
      </dataSource>
    </environment>
  </environments>
  <mappers>
    <mapper resource="member.xml" />
  </mappers>
</configuration>
```

member.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="members">
    <resultMap type="member" id="resultMap">
        <result property="id" column="id" />
        <result property="password" column="password" />
        <result property="name" column="name" />
        <result property="address" column="address" />
        <result property="tel" column="tel" />
        <result property="reg_date" column="reg_date" />
        <result property="del" column="del" />
    </resultMap>
    <insert id="insert" parameterType="member">
        insert into member2 values(#{id}, #{password}, #{name},
            #{address}, #{tel}, sysdate, 'n')
    </insert>
    <select id="select" parameterType="string" resultType="member">
        select * from member2 where id=#{id} and del='n'
    </select>
    <update id="update" parameterType="member">
        update member2 set password=#{password}, name=#{name},
            tel=#{tel}, address=#{address} where id=#{id}
    </update>
    <update id="delete" parameterType="string">
        update member2 set del = 'y' where id = #{id}
    </update>
</mapper>
```

myBatis.model. Member

```
import java.sql.Date;
public class Member {
    private String id;           private String password;
    private String name;        private String address;
    private String tel;         private Date reg_date;
    private String del;
    public String getId() {      return id;          }
    public void setId(String id) { this.id = id;      }
    public String getPassword() { return password; }
    public void setPassword(String password) { this.password=password; }
    public String getName() { return name;          }
    public void setName(String name) { this.name = name;      }
    public String getAddress() { return address;      }
    public void setAddress(String address) { this.address = address; }
    public String getTel() { return tel;              }
    public void setTel(String tel) { this.tel = tel;      }
    public Date getReg_date() { return reg_date;      }
    public void setReg_date(Date reg_date) {this.reg_date = reg_date; }
    public String getDel() { return del;              }
    public void setDel(String del) { this.del = del;      }
}
```


MemberDAO.java

```
package myBatis.dao;
import java.io.IOException;
import java.io.Reader;
import org.apache.ibatis.io.Resources;
import org.apache.ibatis.session.SqlSession;
import org.apache.ibatis.session.SqlSessionFactory;
import org.apache.ibatis.session.SqlSessionFactoryBuilder;
import myBatis.model.Member;
public class MemberDao {
    private static MemberDao instance = new MemberDao();
    private MemberDao() {}
    public static MemberDao getInstance() {    return instance;    }
    private static SqlSession session;
    static {
        try{Reader reader =
            Resources.getResourceAsReader("configuration.xml");
            SqlSessionFactory ssf =
                new SqlSessionFactoryBuilder().build(reader);
            session = ssf.openSession(true);
        } catch (IOException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

MemberDAO.java

```
public int insert(Member member) {
    int result = 0;
    result = session.insert("membersns.insert", member);
    return result;
}
public int loginChk(Member member) {
    int result = -1;
    Member mem = (Member)
    session.selectOne("membersns.select", member.getId());
    if (mem == null) result = -1;
    else {
        String dbPass = mem.getPassword();
        if (dbPass.equals(member.getPassword())) result = 1;
        else result = 0;
    }
    return result;
}
public Member select(String id) {
    Member member = (Member)
        session.selectOne("membersns.select", id);
    return member;
}
```

MemberDAO.java

```
public int update(Member member) {  
    int result = 0;  
    result = session.update("membersns.update", member);  
  
    return result;  
}  
public int delete(String id) {  
    int result = 0;  
    result = session.update("membersns.delete", id);  
  
    return result;  
}  
}
```

command.properties

joinForm.do=myBatis.service.JoinForm
join.do=myBatis.service.JoinAction
loginForm.do=myBatis.service.LoginForm
login.do=myBatis.service.LoginAction
main.do=myBatis.service.MainAction
view.do=myBatis.service.ViewAction
updateForm.do=myBatis.service.UpdateForm
update.do=myBatis.service.UpdateAction
delete.do=myBatis.service.DeleteAction
logout.do=myBatis.service.Logout

index.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!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>Insert title here</title>
</head>
<body>
<script type="text/javascript">
location.href = "main.do";
</script>
</body>
</html>
```

myBatis.service

```
package myBatis.service;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public interface CommandProcess {
    String requestPro(HttpServletRequest request, HttpServletResponse
response);
}
```

```
package myBatis.service;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class JoinForm implements CommandProcess {
    public String requestPro(HttpServletRequest request, HttpServletResponse
response) {
        return "joinForm";
    }
}
```

joinForm.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html> <html> <head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Insert title here</title>
<link rel="stylesheet" type="text/css" href="common.css">
</head> <body> <form action="join.do">
<table> <caption>회원 가입</caption>
    <tr> <th>아이디</th> <td> <input type="text" name="id"
        required="required" autofocus="autofocus"> </td> </tr>
    <tr> <th>암호</th> <td> <input type="password" name="password"
        required="required"> </td> </tr>
    <tr> <th>이름</th> <td> <input type="text" name="name"
        required="required"> </td> </tr>
    <tr> <th>주소</th> <td> <input type="text" name="address"
        required="required"> </td> </tr>
    <tr> <th>전화</th> <td> <input type="tel" name="tel"
        required="required" pattern="\Wd{3}-\Wd{3,4}-\Wd{4}"
        placeholder="xxx-xxxx-xxxx"> </td> </tr>
    <tr> <th colspan="2"> <input type="submit" value="확인"> </th> </tr>
</table> </form> </body> </html>
```

Confirm

```
package myBatis.service;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

import myBatis.dao.MemberDao;
import myBatis.model.Member;
public class Confirm implements CommandProcess {
    public String requestPro(HttpServletRequest request, HttpServletResponse
response) {
        String id = request.getParameter("id");

        MemberDao md = MemberDao.getInstance();
        Member member = md.select(id);
        request.setAttribute("member", member);
        request.setAttribute("id", id);
        return "confirm";
    }
}
```


Confirm

```
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core"%>
<link rel="stylesheet" type="text/css" href="common.css">
<script type="text/javascript">
    function cl() {
        opener.frm.id.value="${id}";
        window.close();
    }
</script> </head> <body>
<c:if test="${not empty member }">
<h2>이미 있는 아이디입니다</h2>
<h2>다른 아이디를 입력하세요</h2>
<form> <table>
    <tr> <th>아이디</th> <td><input type="text" name="id"
        required="required"> </td> </tr>
    <tr> <th colspan="2"><input type="submit" value="확인"> </th> </tr>
</table> </form>
</c:if>
<c:if test="${empty member}">
<h2>사용 가능한 아이디 입니다</h2>
<button onclick="cl()">닫기</button>
</c:if> </body> </html>
```

JoinAction

```
package myBatis.service;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class JoinAction implements CommandProcess {
    public String requestPro(HttpServletRequest request,
        HttpServletResponse response) {
        String id = request.getParameter("id");
        String password = request.getParameter("password");
        String name = request.getParameter("name");
        String address = request.getParameter("address");
        String tel = request.getParameter("tel");
        Member member = new Member();
        member.setId(id);          member.setPassword(password);
        member.setName(name); member.setAddress(address);
        member.setTel(tel);
        MemberDao md = MemberDao.getInstance();
        int result = md.insert(member);
        request.setAttribute("result", result);
        return "join";
    }
}
```

join.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 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>Insert title here</title> </head> <body>
<c:if test="${result > 0 }">
    <script type="text/javascript">
        alert("회원가입 성공");
        location.href="loginForm.do";
    </script>
</c:if>
<c:if test="${result == 0 }">
    <script type="text/javascript">
        alert("애효 !!");
        history.go(-1);
    </script>
</c:if>
</body>
</html>
```

LoginForm

```
package myBatis.service;  
import javax.servlet.http.HttpServletRequest;  
import javax.servlet.http.HttpServletResponse;  
public class LoginForm implements CommandProcess {  
    public String requestPro(HttpServletRequest request,  
        HttpServletResponse response) {  
        return "loginForm";  
    }  
}
```

loginForm.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html> <html> <head> <meta http-equiv="Content-Type"
content="text/html; charset=UTF-8">
<title>Insert title here</title>
<link rel="stylesheet" type="text/css" href="common.css">
</head> <body>
<form action="login.do">
<table> <caption>로그인</caption>
    <tr> <th>아이디</th> <td> <input type="text" name="id"
        required="required" autofocus="autofocus"> </td> </tr>
    <tr> <th>암호</th> <td> <input type="password" name="password"
        required="required"> </td> </tr>
    <tr> <th colspan="2"> <input type="submit" value="확인"> </th> </tr>
</table>
</form>
<a href="joinForm.do"> 회원가입 </a>
</body>
</html>
```

LoginAction

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class LoginAction implements CommandProcess {
    public String requestPro(HttpServletRequest request,
        HttpServletResponse response) {
        String id = request.getParameter("id");
        String password = request.getParameter("password");
        Member member = new Member();
        member.setId(id);
        member.setPassword(password);
        MemberDao md = MemberDao.getInstance();
        int result = md.loginChk(member);
        if (result > 0) {
            HttpSession session = request.getSession();
            session.setAttribute("id", id);
        }
        request.setAttribute("result", result);
        return "login";
    }
}
```

Login.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html> <head> <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8"> <body>
<c:if test="${result > 0 }">
    <script type="text/javascript">
        alert("로그인 성공 방가 !"); location.href = "main.do";
    </script>
</c:if>
<c:if test="${result == 0 }">
    <script type="text/javascript">
        alert("아이디 똑바로"); history.go(-1);
    </script>
</c:if>
<c:if test="${result == -1 }">
    <script type="text/javascript">
        alert("넌 누구냐"); history.go(-1);
    </script>
</c:if>
</body> </html>
```


MainAction

```
package myBatis.service;

import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import myBatis.dao.MemberDao;
import myBatis.model.Member;

public class MainAction implements CommandProcess {
    public String requestPro(HttpServletRequest request,
        HttpServletResponse response) {
        HttpSession session = request.getSession();
        String id = (String)session.getAttribute("id");
        if (id != null ) {
            MemberDao md = MemberDao.getInstance();
            Member member = md.select(id);
            request.setAttribute("member", member);
        }
        return "main";
    }
}
```


sessionChk.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<body>
<c:set var="id" value="${sessionScope['id']}"> </c:set>
<c:if test="${empty id}">
    <script type="text/javascript">
        location.href="loginForm.do";
    </script>
</c:if>
<%-- <%
    String id = (String)session.getAttribute("id");
    if (id == null || id.equals("")) {
        response.sendRedirect("loginForm.do");
        return;
    }
%> --%>
</body>
</html>
```

main.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ include file="sessionChk.jsp" %>
<!DOCTYPE html>
<html> <head> <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
<title>Insert title here</title>
<link rel="stylesheet" type="text/css" href="common.css">
</head> <body>
<h2>${member.name }님 반갑습니다</h2>
<table>
    <tr> <td> <a href="view.do">회원정보 조회</a> </td> </tr>
    <tr> <td> <a href="updateForm.do">회원정보 수정</a> </td> </tr>
    <tr> <td> <a href="delete.do">회원 탈퇴</a> </td> </tr>
    <tr> <td> <a href="logout.do">로그아웃</a> </td> </tr>
</table>
</body>
</html>
```

ViewAction

```
package myBatis.service;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import myBatis.dao.MemberDao;
import myBatis.model.Member;
public class ViewAction implements CommandProcess {
    public String requestPro(HttpServletRequest request, HttpServletResponse
response) {
        HttpSession session = request.getSession();
        String id = (String)session.getAttribute("id");
        MemberDao md = MemberDao.getInstance();
        Member member = md.select(id);
        request.setAttribute("member", member);
        return "view";
    }
}
```

view.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ include file="sessionChk.jsp" %>
<!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>Insert title here</title>
<link rel="stylesheet" type="text/css" href="common.css">
</head><body>
<table><caption>회원정보</caption>
<tr><th>아이디</th><td>${member.id }</td></tr>
<tr><th>이름</th><td>${member.name }</td></tr>
<tr><th>전화</th><td>${member.tel }</td></tr>
<tr><th>주소</th><td>${member.address }</td></tr>
<tr><th>등록일</th><td>${member.reg_date}</td></tr>
</table>
<a href="main.do">메인</a>
<a href="updateForm.do">회원정보 수정</a>
</body>
</html>
```

UpdateForm

```
package myBatis.service;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import myBatis.dao.MemberDao;
import myBatis.model.Member;
public class UpdateForm implements CommandProcess {
    public String requestPro(HttpServletRequest request,
        HttpServletResponse response) {
        HttpSession session = request.getSession();
        String id = (String)session.getAttribute("id");
        MemberDao md = MemberDao.getInstance();
        Member member = md.select(id);
        request.setAttribute("member", member);
        return "updateForm";
    }
}
```

updateForm.jsp

```
<%@ include file="sessionChk.jsp" %>
<script type="text/javascript">
    function chk() {
        if (frm.password.value != frm.password2.value) {
            alert("암호가 암호확인과 다릅니다");
            frm.password.focus(); return false;
        }
    }
</script> </head> <body>
<form action="update.do" name="frm" onsubmit="return chk()">
    <input type="hidden" name="id" value="{member.id}">
    <table> <caption>회원정보</caption>
        <tr> <th>아이디</th> <td>{member.id}</td> </tr>
        <tr> <th>암호</th> <td><input type="password" name="password"
            required="required" autofocus="autofocus"></td> </tr>
        <tr> <th>암호확인</th> <td><input type="password" name="password2"
            required="required"></td> </tr>
        <tr> <th>이름</th> <td><input type="text" name="name"
            required="required" value="{member.name}"></td> </tr>
        <tr> <th>전화</th> <td><input type="tel" name="tel"
            required="required" value="{member.tel}"
            pattern="\d{3}-\d{3,4}-\d{4}"></td> </tr>
        <tr> <th>주소</th> <td><input type="text" name="address"
            required="required" value="{member.address}"></td> </tr>
        <tr> <th>등록일</th> <td>{member.reg_date}</td> </tr>
        <tr> <th colspan="2"><input type="submit" value="확인"> </th> </tr>
    </table>
</form> </body> </html>
```

UpdateAction

```
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class UpdateAction implements CommandProcess {
    public String requestPro(HttpServletRequest request,
        HttpServletResponse response) {
        String id = request.getParameter("id");
        String password = request.getParameter("password");
        String name = request.getParameter("name");
        String tel = request.getParameter("tel");
        String address = request.getParameter("address");
        Member member = new Member();
        member.setId(id);
        member.setPassword(password);
        member.setName(name); member.setTel(tel);
        member.setAddress(address);
        MemberDao md = MemberDao.getInstance();
        int result = md.update(member);
        request.setAttribute("result", result);
        return "update";
    }
}
```


update.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ include file="sessionChk.jsp" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<html> <head> <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8">
<title>Insert title here</title> </head> <body>
<c:if test="${result > 0 }">
    <script type="text/javascript">
        alert("수정 성공 ㅋㅋ");
        location.href="view.do";
    </script>
</c:if>
<c:if test="${result == 0 }">
    <script type="text/javascript">
        alert("수정 실패 ㅠㅠ");
        history.go(-1);
    </script>
</c:if>
</body>
</html>
```


DeleteAction

```
package myBatis.service;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

import myBatis.dao.MemberDao;
public class DeleteAction implements CommandProcess {
    public String requestPro(HttpServletRequest request,
        HttpServletResponse response) {
        HttpSession session = request.getSession();
        String id = (String)session.getAttribute("id");
        MemberDao md = MemberDao.getInstance();
        int result = md.delete(id);
        if (result > 0) {
            session.invalidate();
        }
        request.setAttribute("result", result);
        return "delete";
    }
}
```

delete.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ include file="sessionChk.jsp" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<title>Insert title here</title></head><body>
<c:if test="${result > 0}">
<script type="text/javascript">
alert("탈퇴 ! 잘되나 보자");
location.href = "login.do";
</script>
</c:if>
<c:if test="${result == 0}">
<script type="text/javascript">
alert("헐 ! 탈퇴 안되네 찌드기");
history.go(-1);
</script>
</c:if>
</body>
</html>
```

Logout

```
package myBatis.service;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
public class Logout implements CommandProcess {
    public String requestPro(HttpServletRequest request,
        HttpServletResponse response) {
        HttpSession session = request.getSession();
        session.invalidate();
        return "loginForm";
    }
}
```