

Spring Framework

Chapter11 mybatis



경민대학교
융합소프트웨어과
박남일(niparkster@gmail.com)

- ▶ mybatis의 개념
- ▶ 스프링 트랜잭션 사용 원리
- ▶ JDBC 설정
- ▶ JDBC 이용

▶ mybatis란?

- 개발자가 지정한 SQL, 저장 프로시저를 지원하는 프레임워크
- 프로그램 소스 안에 SQL 문 작성(기존방식) → SQL문을 프로그램에서 분리 XML 파일에 작성

▶ mybatis 장점

- 코딩량 절감
- 간편한 유지보수(기존 프로그램 수정 X → XML 파일의 SQL문만 변경)

▶ mybatis 역사

- ibatis 라는 이름으로 2.5까지 개발 (아파치 프로젝트)
- 3.0 version 이후로 mybatis 로 이름 변경

▶ 로깅툴

- 로깅툴을 사용하는 이유

- System.out.println() 명령어는 IO 리소스를 많이 사용 → 시스템 느려짐
- 로그를 파일로 저장하여 분석 필요성

- 로깅툴 종류

- commons-logging (과거버전)
- log4j : 효율적 메모리 관리로 많이 사용 (과거버전)
- logback : log4j 보다 성능이 더 우수 (최근사용)
→ slf4j : logback을 사용하기 위한 인터페이스

▶ slf4j 설정

- pom.xml의 slf4j-version을 1.7.25로 설정

```
<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/maven-v4_0_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.spring11</groupId>
  <artifactId>ex</artifactId>
  <name>spring11_1</name>
  <packaging>war</packaging>
  <version>1.0.0-BUILD-SNAPSHOT</version>
  <properties>
    <java-version>1.8</java-version>
    <org.springframework-version>5.1.3.RELEASE</org.springframework-version>
    <org.aspectj-version>1.9.2</org.aspectj-version>
    <org.slf4j-version>1.7.25</org.slf4j-version>
  </properties>
```

▶ slf4j 설정

- pom.xml에 라이브러리(dependency) 추가

```
<dependency>
  <groupId>ch.qos.logback</groupId>
  <artifactId>logback-classic</artifactId>
  <version>1.2.3</version>
</dependency>
```

```
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>slf4j-api</artifactId>
  <version>${org.slf4j-version}</version>
</dependency>
```

```
<dependency>
  <groupId>org.slf4j</groupId>
  <artifactId>jcl-over-slf4j</artifactId>
  <version>1.8.0-beta4</version>
</dependency>
```

```
<dependency>
  <groupId>uk.co.nicholutions</groupId>
  <artifactId>slf4j-log4j12</artifactId>
  <version>1.7.22-CUSTOM</version>
</dependency>
```

```
<dependency>
  <groupId>log4j</groupId>
  <artifactId>log4j</artifactId>
  <version>1.2.17</version>
  <exclusions>
    <exclusion>
      <groupId>javax.mail</groupId>
      <artifactId>mail</artifactId>
    </exclusion>
    <exclusion>
      <groupId>javax.jms</groupId>
      <artifactId>jms</artifactId>
    </exclusion>
    <exclusion>
      <groupId>com.sun.jdmk</groupId>
      <artifactId>jmxtools</artifactId>
    </exclusion>
    <exclusion>
      <groupId>com.sun.jmx</groupId>
      <artifactId>jmxri</artifactId>
    </exclusion>
  </exclusions>
  <scope>runtime</scope>
</dependency>
```

▶ slf4j 설정

- pom.xml에 라이브러리(dependency) 추가

```
<!-- mybatis-spring -->
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis-spring</artifactId>
  <version>1.2.3</version>
</dependency>

<!-- mybatis -->
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis</artifactId>
  <version>3.3.0</version>
</dependency>

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

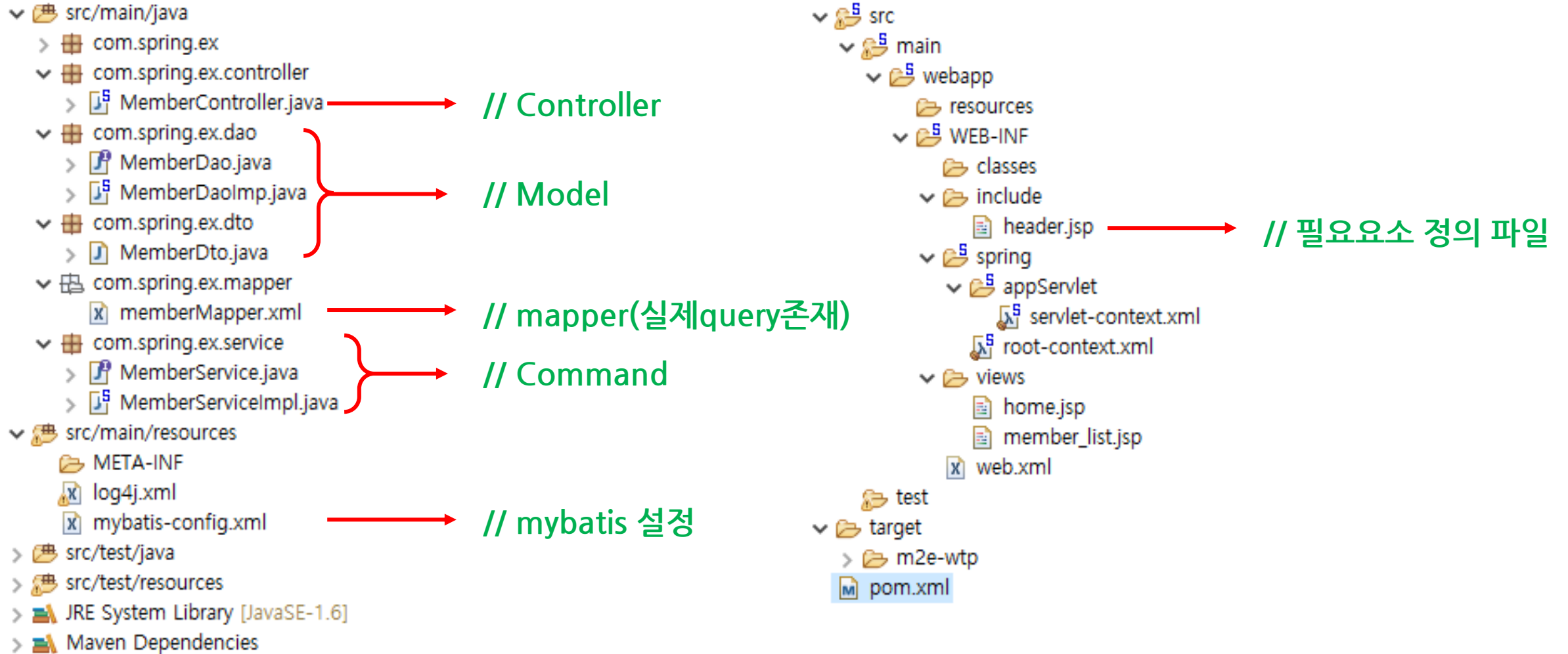
<!-- ojdbc6 -->
<dependency>
  <groupId>com.oracle</groupId>
  <artifactId>ojdbc6</artifactId>
  <version>11.2.0.3</version>
</dependency>
```

▶ 실습을 위한 테이블 생성 및 테스트 데이터 삽입

```
create table member (  
  id varchar2(50) not null primary key,  
  pw varchar2(50) not null,  
  name varchar2(50),  
  email VARCHAR2(50),  
  join_date date default sysdate  
);  
  
insert into member(id, pw, name, email) values ('aaa', '111', '박남일', 'a@a.com');  
  
commit;
```

| | ID | PW | NAME | EMAIL | JOIN_DATE |
|---|-----|-----|------|---------|-----------|
| 1 | aaa | 111 | 박남일 | a@a.com | 19/05/15 |

▶ 프로젝트 구조도



▶ 프로젝트 구조도

- src/main/java
 - com.spring.ex
 - com.spring.ex.controller
 - MemberController.java
 - com.spring.ex.dao
 - MemberDao.java
 - MemberDaoImp.java
 - com.spring.ex.dto
 - MemberDto.java
 - com.spring.ex.mapper
 - memberMapper.xml
 - com.spring.ex.service
 - MemberService.java
 - MemberServiceImpl.java
- src/main/resources
 - META-INF
 - log4j.xml
 - mybatis-config.xml
- src/test/java
- src/test/resources
- JRE System Library [JavaSE-1.6]
- Maven Dependencies

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- xml 지시어 -->
<!DOCTYPE configuration
PUBLIC "-//mybatis.org//DTD config 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
  <!-- 알리아스 설정(경로 별명) -->
  <typeAliases>
    <!-- typeAlias type= "전체경로" alias= "별칭" -->
  </typeAliases>
  <!-- 유 연결 참조코드 -->
  <environments default= " " >
    <environment id= " " >
      <transactionManager type= "JDBC" />
      <dataSource type= "POOLED">
        <property name= "driver" value= "oracle.jdbc.driver.OracleDriver" />
        <property name= "url" value= "jdbc:oracle:thin:@localhost:1521:xe" />
        <property name= "username" value= "clara" />
        <property name= "password" value= "agnes" />
      </dataSource>
    </environment>
  </environments>
  <!-- 실제 sql query -->
  <mappers>
    <mapper resource= "memo/mapper/memo.xml" />
  </mappers>
</configuration>
```

// Configuration 지시 세팅 부분

// 경로 별칭 세팅

// JDBC 연결(DBCP 사용 가능)

// mapper(query 존재) 설정

▶ 프로젝트 구조도

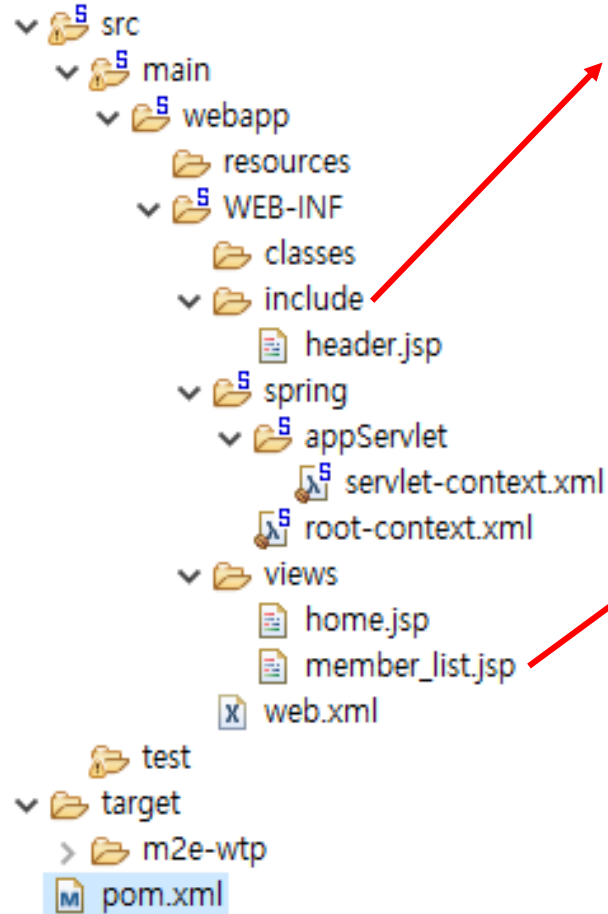
- ▼ src/main/java
 - > com.spring.ex
 - ▼ com.spring.ex.controller
 - > MemberController.java
 - ▼ com.spring.ex.dao
 - > MemberDao.java
 - > MemberDaoImp.java
 - ▼ com.spring.ex.dto
 - > MemberDto.java
 - ▼ com.spring.ex.mapper
 - memberMapper.xml
 - ▼ com.spring.ex.service
 - > MemberService.java
 - > MemberServiceImpl.java
- ▼ src/main/resources
 - META-INF
 - log4j.xml
 - mybatis-config.xml
- > src/test/java
- > src/test/resources
- > JRE System Library [JavaSE-1.6]
- > Maven Dependencies

```
public class MemberDto {  
    private String id;  
    private String pw;  
    private String name;  
    private String email;  
    private Date join_date;  
}
```

// getters & setters 존재

```
public MemberDto() {  
  
}  
@Override  
public String toString() {  
    return "MemberDto [id=" + id + ", pw=" + pw + ", name=" + name + ", email=" + email  
        + ", join_date=" + join_date + "];"  
}  
}
```

▶ 프로젝트 구조도



// viewer의 header부분

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
<c:set var="path" value="${pageContext.request.contextPath}" />

<script src="http://code.jquery.com/jquery-3.3.1.min.js"></script>
```

```
<%@ include file="../include/header.jsp" %>
</head>
<body>
    <h1>회원목록</h1>
    <c:forEach var="row" items="${list}">
        ${row.id}
        ${row.name}
        ${row.email}
        ${row.join_date}
    </c:forEach>
</body>
</html>
```

▶ 프로젝트 구조도

- ▼ src/main/java
 - > com.spring.ex
 - ▼ com.spring.ex.controller
 - > MemberController.java
 - ▼ com.spring.ex.dao
 - > MemberDao.java
 - > MemberDaoImpl.java
 - ▼ com.spring.ex.dto
 - > MemberDto.java
 - ▼ com.spring.ex.mapper
 - > memberMapper.xml
 - ▼ com.spring.ex.service
 - > MemberService.java
 - > MemberServiceImpl.java
- ▼ src/main/resources
 - > META-INF
 - > log4j.xml
 - > mybatis-config.xml
- > src/test/java
- > src/test/resources
- > JRE System Library [JavaSE-1.6]
- > Maven Dependencies

```
@Controller // Controller 자동 생성
public class MemberController {
    @Inject // new 연산자로 생성 필요 없이 자동 주입
    MemberService memberService;

    @RequestMapping("list.do")
    public String memberList(Model model) {
        System.out.println("memberService : " + memberService);
        List<MemberDto> list = memberService.memberList();
        model.addAttribute("list", list);
        return "member_list";
    }
}
```

▶ 프로젝트 구조도

- src/main/java
 - com.spring.ex
 - com.spring.ex.controller
 - MemberController.java
 - com.spring.ex.dao
 - MemberDao.java
 - MemberDaoImpl.java
 - com.spring.ex.dto
 - MemberDto.java
 - com.spring.ex.mapper
 - memberMapper.xml
 - com.spring.ex.service
 - MemberService.java
 - MemberServiceImpl.java
- src/main/resources
 - META-INF
 - log4j.xml
 - mybatis-config.xml
- src/test/java
- src/test/resources
- JRE System Library [JavaSE-1.6]
- Maven Dependencies

```
public interface MemberService { // command interface
    public List<MemberDto> memberList();
    public void insertMember(MemberDto dto);
    public MemberDto viewMember(String id);
    public void deleteMember(String id);
    public void updateMember(MemberDto dto);
    public boolean chPw(String id, String pw);
}
```

```
@Service // Service 자동 생성
public class MemberServiceImpl implements MemberService {
    @Inject // new 연산자로 생성 필요 없이 자동 주입
    MemberDao memberDao;
    @Override
    public List<MemberDto> memberList() {
        return memberDao.memberList();
    }
    @Override
    public void insertMember(MemberDto dto) {
    }
    @Override
    public MemberDto viewMember(String id) {
        return null;
    }
}
```

// 밑으로 인터페이스의 메서드 override

▶ 프로젝트 구조도

- ▼ src/main/java
 - > com.spring.ex
 - ▼ com.spring.ex.controller
 - > MemberController.java
 - ▼ com.spring.ex.dao
 - > MemberDao.java
 - > MemberDaoImp.java
 - ▼ com.spring.ex.dto
 - > MemberDto.java
 - ▼ com.spring.ex.mapper
 - > memberMapper.xml
 - ▼ com.spring.ex.service
 - > MemberService.java
 - > MemberServiceImpl.java
- ▼ src/main/resources
 - META-INF
 - log4j.xml
 - mybatis-config.xml
- > src/test/java
- > src/test/resources
- > JRE System Library [JavaSE-1.6]
- > Maven Dependencies

```

public interface MemberDao {           // DAO interface
    public List<MemberDto> memberList();
    public void insertMember(MemberDto dto);
    public MemberDto viewMember(String id);
    public void deleteMember(String id);
    public void updateMember(MemberDto dto);
    public boolean checkPw(String id, String pw);
}

// 서버가 시작 될 때 이 클래스가 자동으로 등록
// (스프링이 해주어서 따로 new를 사용 X)
@Repository
public class MemberDaoImp implements MemberDao {
    @Inject // new 연산자로 생성 필요 없이 자동 주입
    SqlSession sqlSession;
    @Override
    public List<MemberDto> memberList() {
        return sqlSession.selectList("member.memberList");
    }
    @Override
    public void insertMember(MemberDto dto) {
        // 밑으로 인터페이스의 메서드 override

        /**
         * mybatis에서는 리턴타입을 ArrayList가 아닌 List로 해야 함
         * SqlSession : mybatis로 sql을 실행시키는 객체
         * SqlSession을 이용하여 member 네이스페이스의 listAll 쿼리를 실행시킴
         */
    }
}
    
```

▶ 프로젝트 구조도

- ▼ src/main/java
 - > com.spring.ex
 - ▼ com.spring.ex.controller
 - > MemberController.java
 - ▼ com.spring.ex.dao
 - > MemberDao.java
 - > MemberDaoImp.java
 - ▼ com.spring.ex.dto
 - > MemberDto.java
 - ▼ com.spring.ex.mapper
 - memberMapper.xml
 - ▼ com.spring.ex.service
 - > MemberService.java
 - > MemberServiceImpl.java
- ▼ src/main/resources
 - META-INF
 - log4j.xml
 - mybatis-config.xml
- > src/test/java
- > src/test/resources
- > JRE System Library [JavaSE-1.6]
- > Maven Dependencies

```
<?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">
```

```
<!-- 다른 mapper와 중복되지 않도록 네임스페이스 기재 -->
<mapper namespace="member">
<select id="memberList" resultType="com.spring.ex.dto.MemberDto">
    select * from member
    order by name
</select>
</mapper>
```

namespace : query 묶음 단위

id : 실행 query 이름(객체 이름)

resultType : query 처리 후 데이터를 담는 객체

10장. 스프링JDBC

수고하셨습니다.
질문 있습니까?

NEXT Chapter

11장. 트랜잭션