



# Mybatis 프레임워크 시작하기



# BoardDAO 비교하기 – JDBC vs. MyBatis

```
public class BoardDAO {
    // JDBC 관련 변수
    private Connection conn = null;
    private PreparedStatement pstmt = null;
    private ResultSet rset = null;

    private final String BOARD_LIST = "select * from board order by seq desc";

    // 글 목록 조회
    public List<BoardVO> getBoardList(BoardVO vo) {
        System.out.println("==> JDBC로 getBoardList() 기능 처리");
        List<BoardVO> boardList = new ArrayList<BoardVO>();
        try {
            conn = JDBCUtil.getConnection();
            pstmt = conn.prepareStatement(BOARD_LIST);
            rset = pstmt.executeQuery();
            while (rset.next()) {
                BoardVO board = new BoardVO();
                board.setSeq(rset.getInt("SEQ"));
                board.setTitle(rset.getString("TITLE"));
                board.setWriter(rset.getString("WRITER"));
                board.setContent(rset.getString("CONTENT"));
                board.setRegDate(rset.getDate("REGDATE"));
                board.setCnt(rset.getInt("CNT"));
                boardList.add(board);
            }
        } catch (Exception e) {
            e.printStackTrace();
        } finally {
            JDBCUtil.close(rset, pstmt, conn);
        }
        return boardList;
    }
}
```

JDBC

```
public class BoardDAO {  
    public List<BoardVO> getBoardList(BoardVO vo) {  
        SqlSession mybatis = SqlSessionFactoryBean.getSqlSessionInstance();  
        return mybatis.selectList("BoardDAO.getBoardList", vo);  
    }  
}
```

# MyBatis

# 1.1 Mybatis 프레임워크 2가지 특징

- **한 두 줄의 자바 코드로 DB 연동을 처리한다는 것**

- XML 파일에 저장된 SQL 명령어를 대신 실행하고 실행 결과를 VO 자바 객체에 자동으로 매핑시켜 줌. ➔ 데이터매퍼(Data Mapper)라고 함

- **SQL 명령어를 자바 코드에서 분리하여 XML파일에 따로 관리한다는 것**

- 만약, SQL 명령어가 DAO 같은 자바 클래스에 저장되면 SQL 명령어만 수정하는 상황에서도 자바 클래스를 다시 컴파일해야 한다.
- SQL 명령어들을 한 곳에 모아서 관리하기도 쉽지 않다.
- 결국, SQL 명령어에 대한 통합 관리를 위해서라도 자바 소스에서 SQL을 분리하는 게 매우 중요함.



# XML파일 분리의 예

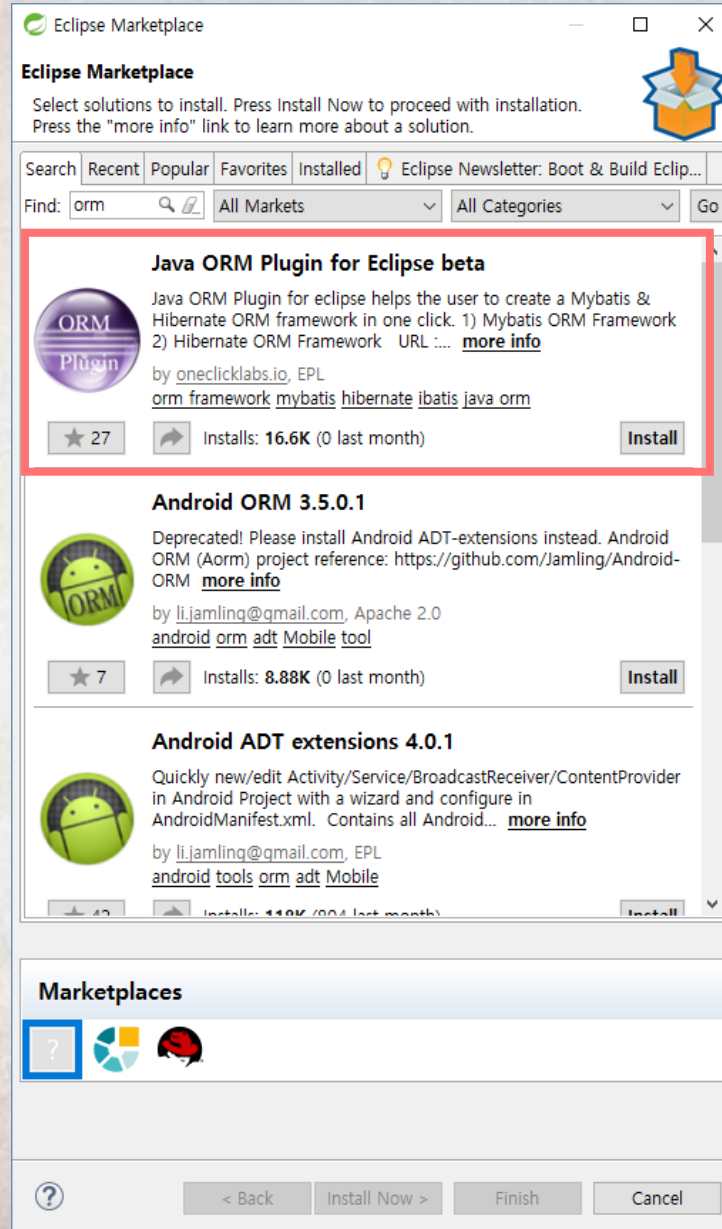
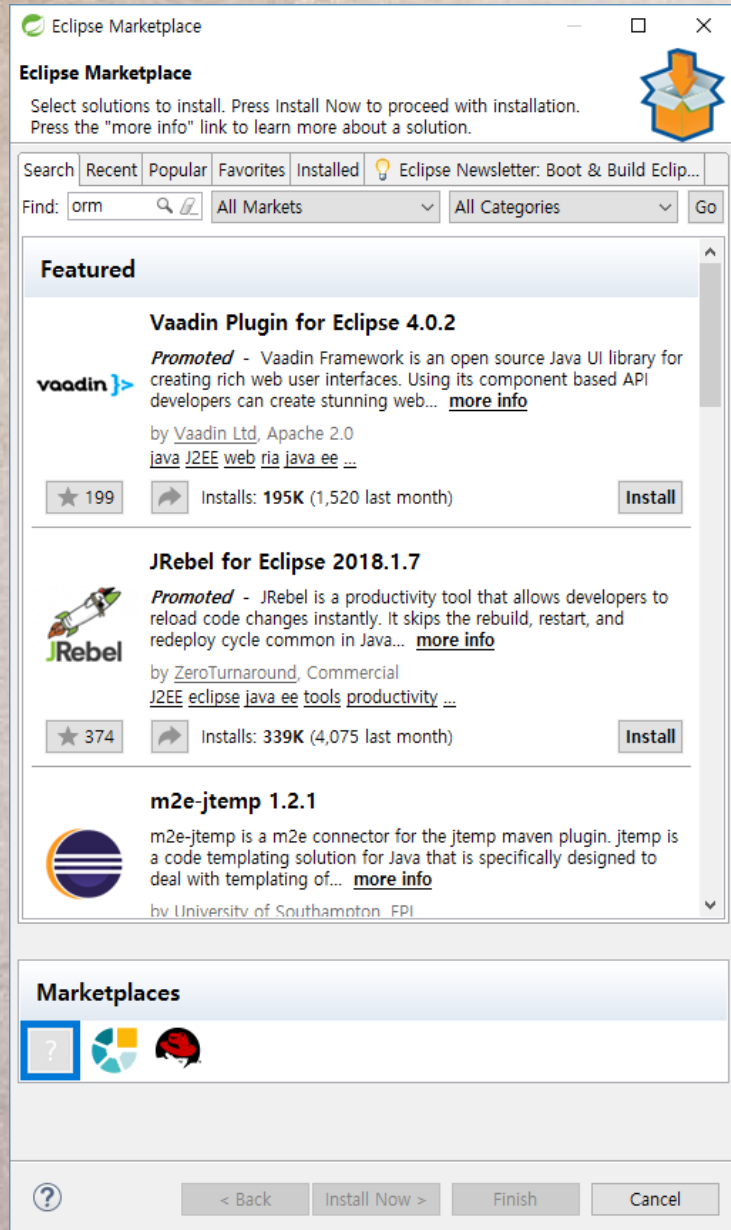
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="kr.ac.inje.comsi.board.BoardVO">

  <select id="getBoardList" resultType="kr.ac.inje.comsi.board.BoardVO" >
    select * from board
  </select>

</mapper>
```

# 1.2 ORM 플러그인 설치 → 제대로 다운로드가 안 됨 ㅠ ㅠ



# 1.3 라이브러리 추가 – Maven Dependency(pom.xml파일)

The screenshot shows the Maven IDE interface with the 'Dependencies' tab selected. The 'BroadWeb/pom.xml' file is open. The 'Dependencies' list on the left includes:

- spring-context : \${org.springframework-version}
- spring-webmvc : \${org.springframework-version}
- aspectjweaver : \${org.aspectj-version}
- slf4j-api : \${org.slf4j-version}
- jcl-over-slf4j : \${org.slf4j-version} [runtime]
- slf4j-log4j12 : \${org.slf4j-version} [runtime]
- log4j : 1.2.15 [runtime]
- javax.inject : 1
- servlet-api : 2.5 [provided]
- jsp-api : 2.1 [provided]
- jstl : 1.2
- junit : 4.7 [test]
- h2 : 1.4.197
- commons-dbcp2 : 2.1.1
- spring-jdbc : 4.3.15.RELEASE

The 'Add...' button is highlighted with a red box. The 'Dependency Management' tab on the right is empty. The bottom status bar shows the following tabs: Overview, Dependencies, Dependency Hierarchy, Effective POM, pom.xml.

To manage your transitive dependency exclusions, please use the [Dependency Hierarchy](#) page.



Select Dependency

Group Id: \*org.mybatis

Artifact Id: \*mybatis

Version: 3.4.1

Scope: compile

Enter groupId, artifactId or sha1 prefix or pattern (\*):

Index downloads are disabled, search results may be incomplete.

Search Results:

?

OK Cancel

Group Id: org.mybatis  
Artifact Id: mybatis  
Version: 3.4.1

Select Dependency

Group Id: \*

Artifact Id: \*

Version:  Scope:

Enter groupId, artifactId or sha1 prefix or pattern (\*):

⚠ Index downloads are disabled, search results may be incomplete.

Search Results:

?

OK Cancel

Group Id: org.mybatis  
Artifact Id: mybatis-spring  
Version: 1.3.0



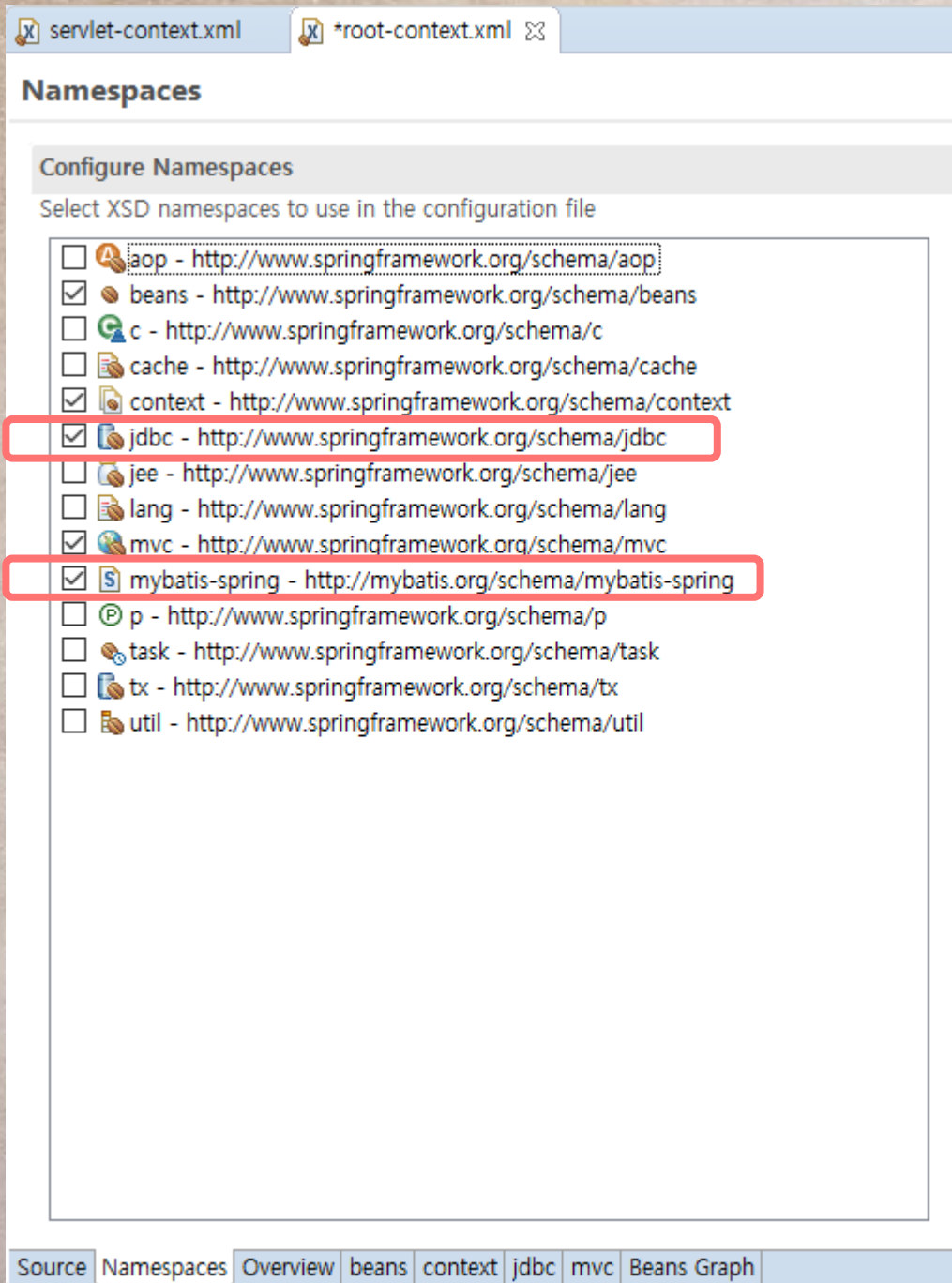
> src/test/resources

▼ Maven Dependencies

- > spring-context-4.3.15.RELEASE.jar - C:\Users\Kyungtae\m2\repository\org\springframework\spring-context\4.3.15.RELEASE.jar
- > spring-aop-4.3.15.RELEASE.jar - C:\Users\Kyungtae\m2\repository\org\springframework\spring-aop\4.3.15.RELEASE.jar
- > spring-beans-4.3.15.RELEASE.jar - C:\Users\Kyungtae\m2\repository\org\springframework\spring-beans\4.3.15.RELEASE.jar
- > spring-core-4.3.15.RELEASE.jar - C:\Users\Kyungtae\m2\repository\org\springframework\spring-core\4.3.15.RELEASE.jar
- > spring-expression-4.3.15.RELEASE.jar - C:\Users\Kyungtae\m2\repository\org\springframework\spring-expression\4.3.15.RELEASE.jar
- > spring-webmvc-4.3.15.RELEASE.jar - C:\Users\Kyungtae\m2\repository\org\springframework\spring-webmvc\4.3.15.RELEASE.jar
- > spring-web-4.3.15.RELEASE.jar - C:\Users\Kyungtae\m2\repository\org\springframework\spring-web\4.3.15.RELEASE.jar
- > aspectjweaver-1.6.10.jar - C:\Users\Kyungtae\m2\repository\org\aspectj\aspectjweaver\1.6.10.jar
- > slf4j-api-1.6.6.jar - C:\Users\Kyungtae\m2\repository\org\slf4j\slf4j-api\1.6.6.jar
- > jcl-over-slf4j-1.6.6.jar - C:\Users\Kyungtae\m2\repository\org\slf4j\jcl-over-slf4j\1.6.6.jar
- > slf4j-log4j12-1.6.6.jar - C:\Users\Kyungtae\m2\repository\org\slf4j\slf4j-log4j12\1.6.6.jar
- > log4j-1.2.15.jar - C:\Users\Kyungtae\m2\repository\log4j\log4j\1.2.15.jar
- > javax.inject-1.jar - C:\Users\Kyungtae\m2\repository\javax\inject\javax.inject\1.jar
- > servlet-api-2.5.jar - C:\Users\Kyungtae\m2\repository\javax\servlet\servlet-api\2.5.jar
- > jsp-api-2.1.jar - C:\Users\Kyungtae\m2\repository\javax\servlet\jsp\jsp-api\2.1.jar
- > jstl-1.2.jar - C:\Users\Kyungtae\m2\repository\javax\servlet\jstl\jstl\1.2.jar
- > junit-4.7.jar - C:\Users\Kyungtae\m2\repository\junit\junit\4.7.jar
- > h2-1.4.197.jar - C:\Users\Kyungtae\m2\repository\com\h2database\h2\1.4.197.jar
- > commons-dbcp2-2.1.1.jar - C:\Users\Kyungtae\m2\repository\org\apache\commons\commons-dbcp2\2.1.1.jar
- > commons-pool2-2.4.2.jar - C:\Users\Kyungtae\m2\repository\org\apache\commons\commons-pool2\2.4.2.jar
- > commons-logging-1.2.jar - C:\Users\Kyungtae\m2\repository\commons-logging\commons-logging\1.2.jar
- > spring-jdbc-4.3.15.RELEASE.jar - C:\Users\Kyungtae\m2\repository\org\springframework\spring-jdbc\4.3.15.RELEASE.jar
- > spring-tx-4.3.15.RELEASE.jar - C:\Users\Kyungtae\m2\repository\org\springframework\spring-tx\4.3.15.RELEASE.jar
- > mybatis-3.4.1.jar - C:\Users\Kyungtae\m2\repository\org\mybatis\mybatis\3.4.1.jar
- > mybatis-spring-1.3.0.jar - C:\Users\Kyungtae\m2\repository\org\mybatis\mybatis-spring\1.3.0.jar

> JRE System Library [JavaSE-1.6]

mybatis-3.4.1.jar  
mybatis-spring-1.3.0.jar



# root-context.xml 파일

- Namespace 항목 추가하기
  - jdbc
  - mybatis-spring



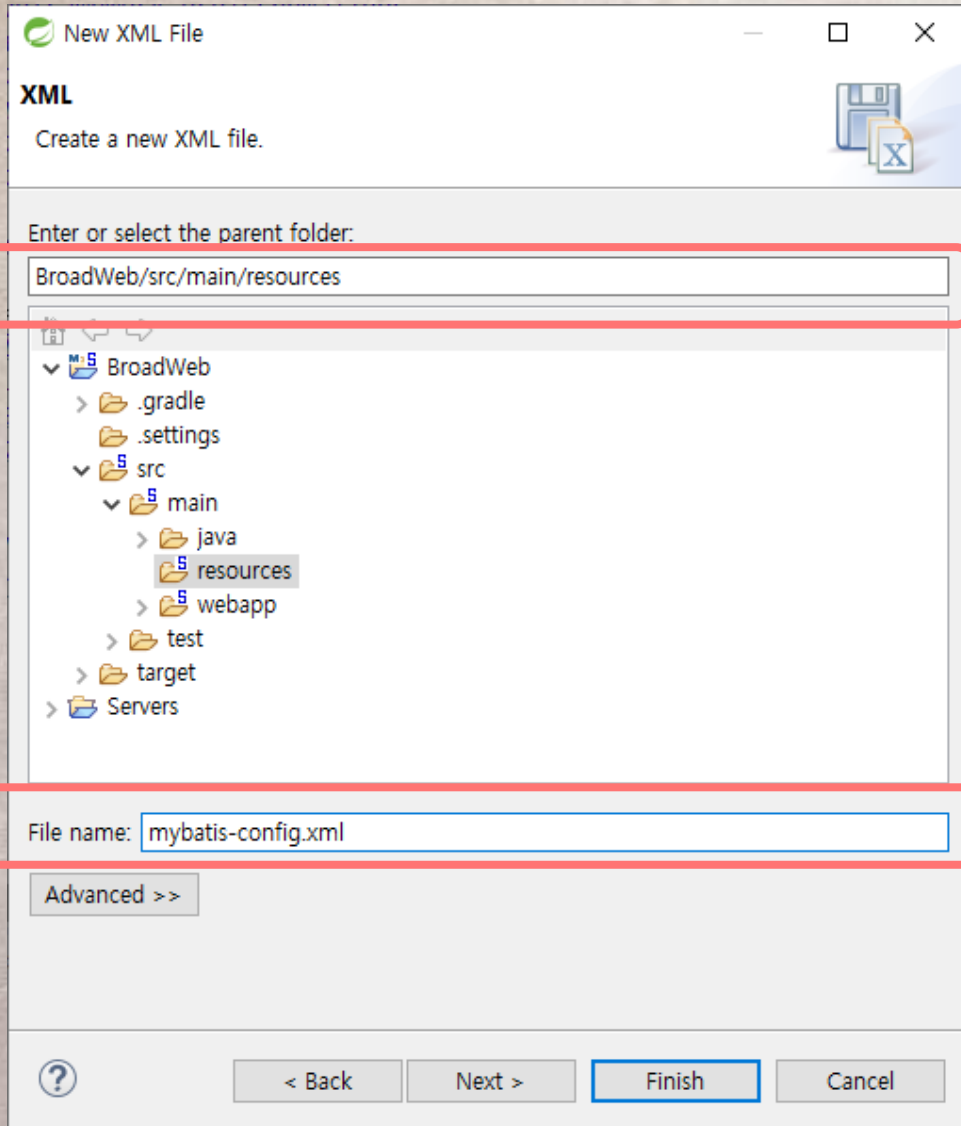
# 1.4 root-context.xml 수정-DataSource 추석 추가

```
servlet-context.xml root-context.xml applicationContext.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:context="http://www.springframework.org/schema/context"
5     xmlns:mvc="http://www.springframework.org/schema/mvc"
6     xmlns:mybatis-spring="http://mybatis.org/schema/mybatis-spring"
7     xmlns:jdbc="http://www.springframework.org/schema/jdbc"
8     xsi:schemaLocation="http://www.springframework.org/schema/jdbc http://www.springframework.org/schema/jdbc/spring-jdbc-4.3.xsd
9         http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd
10        http://mybatis.org/schema/mybatis-spring http://mybatis.org/schema/mybatis-spring-1.2.xsd
11        http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.3.xsd
12        http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd">
13
14     <context:component-scan base-package="kr.ac.inje.comsi.board"></context:component-scan>
15
16     <bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
17         <property name="driverClassName" value="org.h2.Driver"></property>
18         <property name="url" value="jdbc:h2:D:/workspace/h2db/db"></property>
19         <property name="username" value="sa"></property>
20         <property name="password" value=""></property>
21     </bean>
22
23 </beans>
24
25
```

주석 처리하기(삭제)

# 1.5 SQL Mapping 프레임워크

- MyBatis는 SQL Mapping 프레임워크로 별도의 설정파일을 가짐



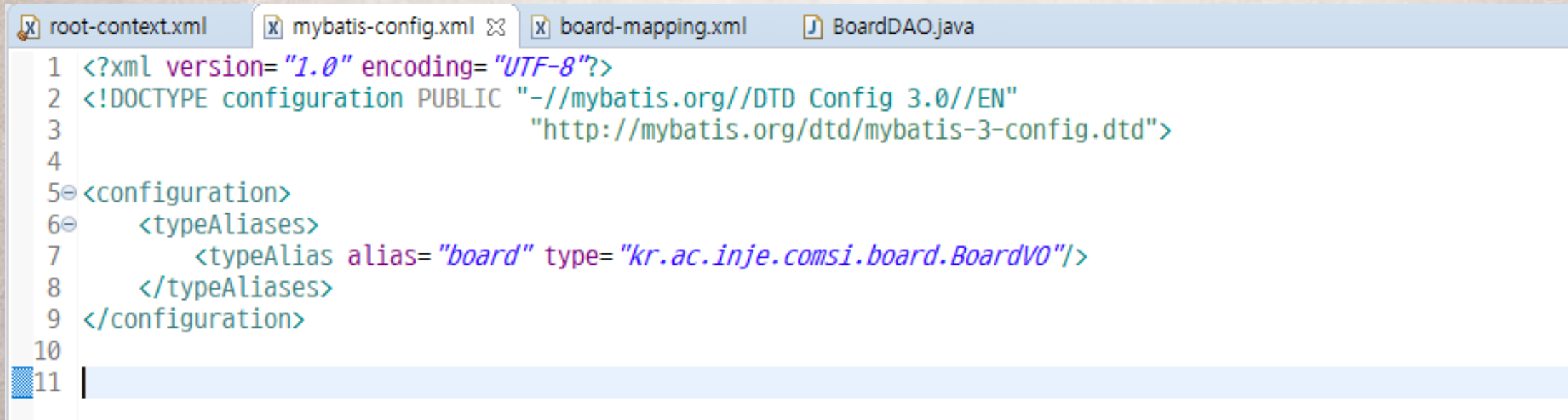
BroadWeb/src/main/resources

mybatis-config.xml



# mybatis-config.xml 파일 수정 – typeAlias 설정

- <typeAliases> 엘리먼트는 <typeAlias>를 여러 개 가질 수 있으며,
- <typeAlias>를 사용하여 특정 클래스의 별칭(alias)를 선언할 수 있다.
- “kr.ac.inje.comsi.board.BoardVO” 클래스를 “board” 별칭으로 선언
- \*-mapping.xml 파일에서 접근 사용가능



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
3     "http://mybatis.org/dtd/mybatis-3-config.dtd">
4
5 <configuration>
6   <typeAliases>
7     <typeAlias alias="board" type="kr.ac.inje.comsi.board.BoardVO"/>
8   </typeAliases>
9 </configuration>
10
11 |
```

# mybatis-config.xml 내용 파일 추가

```
root-context.xml | mybatis-config.xml | board-mapping.xml | BoardDAO.java | SqlSessionFactoryBean.java | BoardServiceClient.java
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
3     "http://mybatis.org/dtd/mybatis-3-config.dtd">
4
5 <configuration>
6     <!-- <properties resource="db.properties"/> -->
7     <typeAliases>
8         <typeAlias alias="board" type="kr.ac.inje.comsi.board.BoardVO"/>
9     </typeAliases>
10    <environments default="development">
11        <environment id="development">
12            <transactionManager type="JDBC"/>
13            <dataSource type="POOLED">
14                <property name="driver" value="org.h2.Driver"/>
15                <property name="url" value="jdbc:h2:D:/workspace/h2db/db"/>
16                <property name="username" value="sa"/>
17                <property name="password" value=""/>
18            </dataSource>
19        </environment>
20    </environments>
21
22    <mappers>
23        <mapper resource="mapper/board-mapping.xml"/>
24    </mappers>
25 </configuration>
26
```

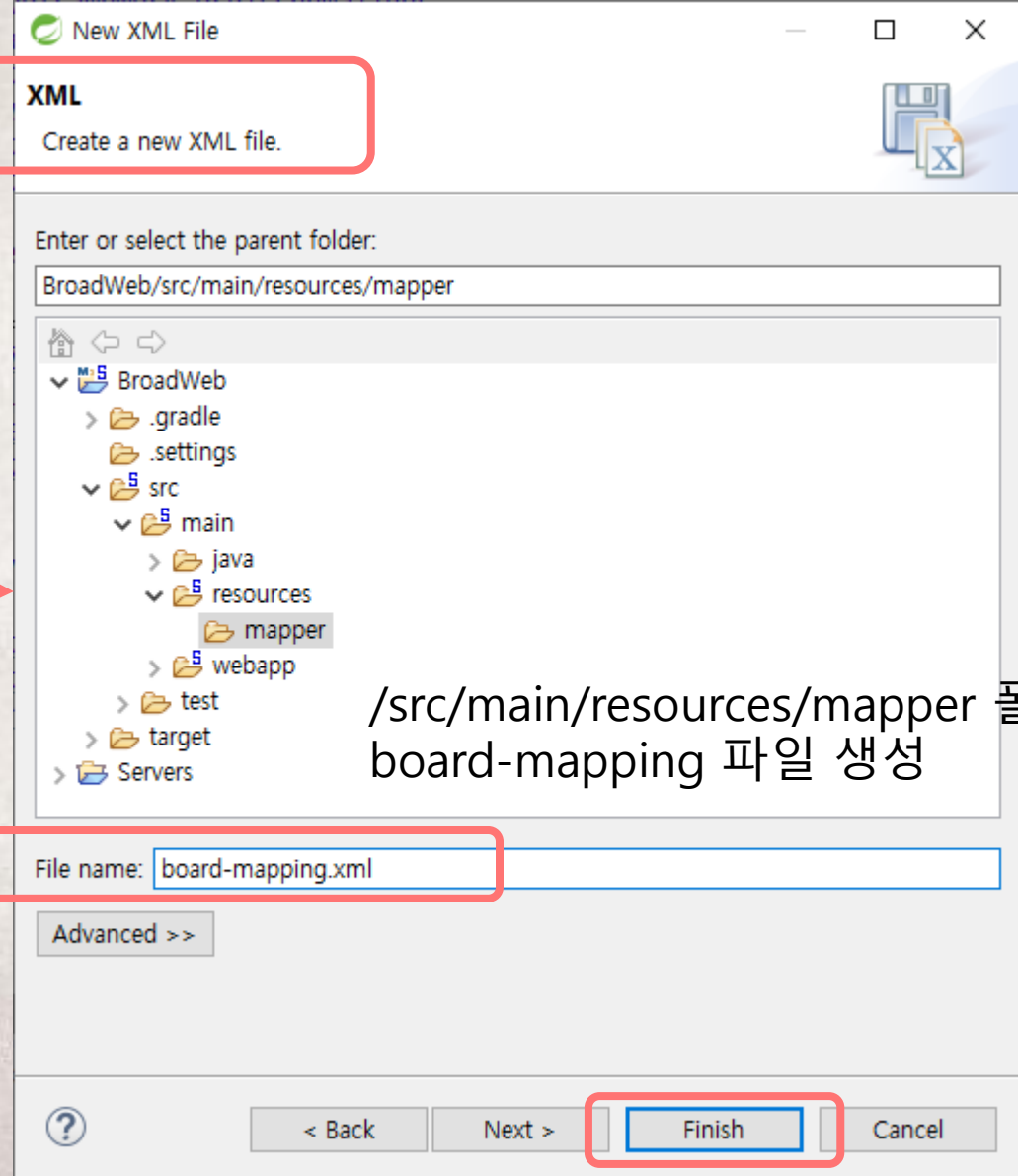
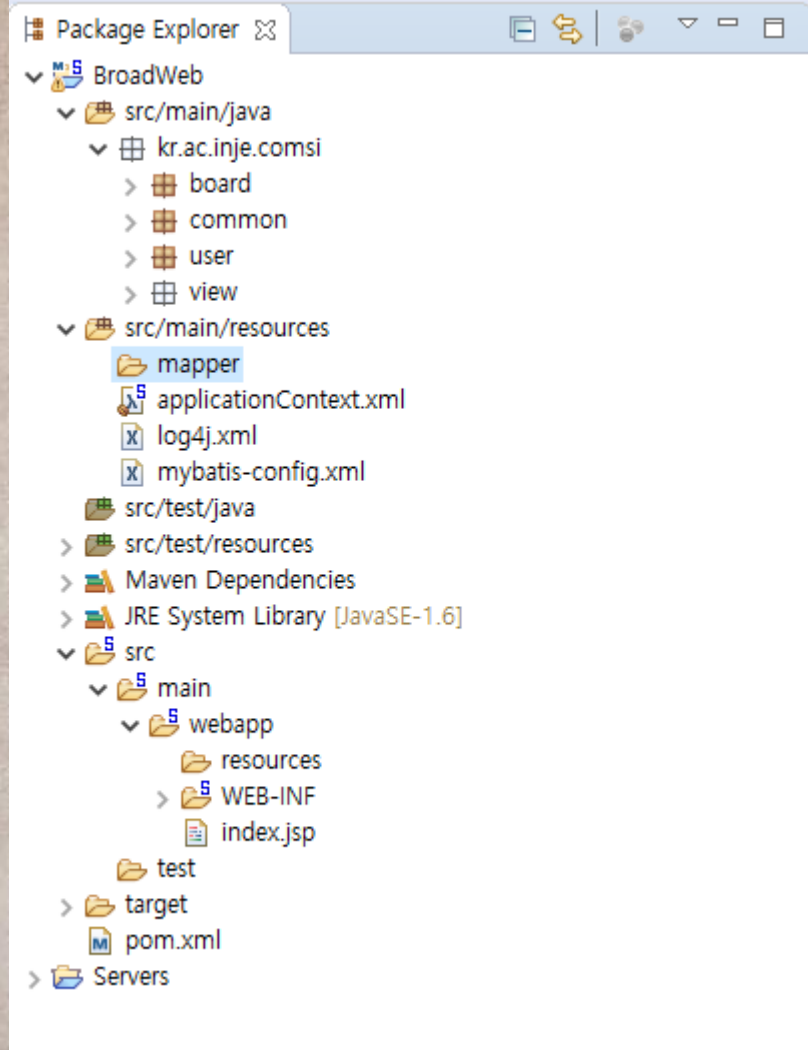
SQL mapper 파일에서 호출 사용

sql mapper 파일 위치 설정

➔ <http://www.mybatis.org/mybatis-3/ko/getting-started.html>

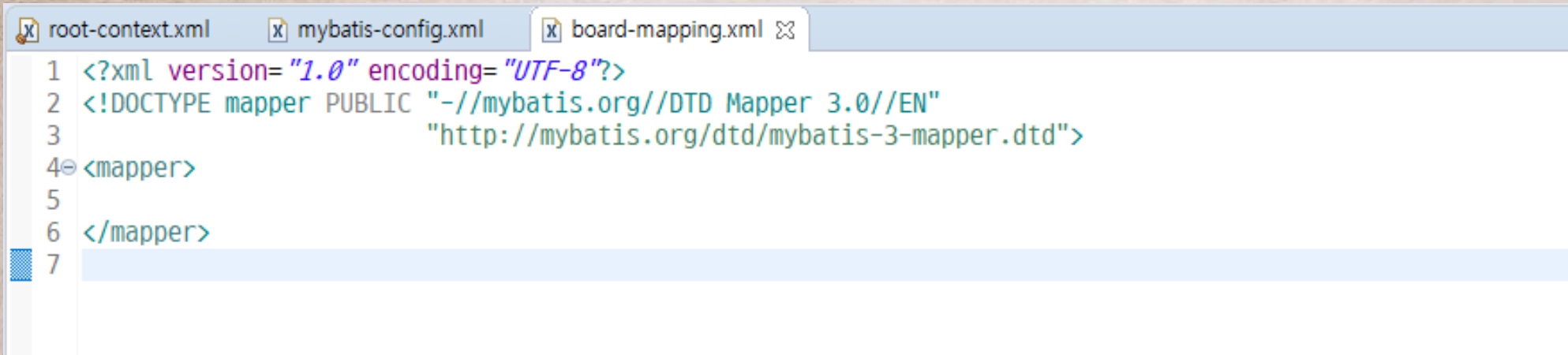


# 1.6 SQL Mapper 파일 설정 – Board Table



/src/main/resources/mapper 폴더 안에  
board-mapping 파일 생성

# board-mapping.xml의 DTD 및 <mapper>root 엘리먼트 설정



```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
3     "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
4 <mapper>
5
6 </mapper>
7
```

# board-mapping.xml 내용 파일 추가

```
root-context.xml  mybatis-config.xml  board-mapping.xml  BoardDAO.java  SqlSessionFactoryBean.java  BoardServiceClient.java

1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
3      "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
4  <mapper namespace="BoardDAO">
5      <insert id="insertBoard">
6          insert into board(seq, title, writer, content)
7              values((select nvl(max(seq),0)+1 from board),#{title}, #{writer},#{content})
8      </insert>
9
10     <select id="getBoard" resultType="board">
11         select * from board where seq=#{seq}
12     </select>
13
14     <select id="getBoardList" resultType="board">
15         select * from board order by seq desc
16     </select>
17
18     <delete id="deleteBoard">
19         delete board where seq=#{seq}
20     </delete>
21
22     <update id="updateBoard">
23         update board set title=#{title}, content=#{content} where seq=#{seq}
24     </update>
25 </mapper>
26
```



## 1.7 SqlSession객체 생성하기-SqlSessionFactoryBean 클래스

- Mybatis에서 DAO를 구현하려면 SqlSession 객체가 필요
- SqlSession 객체를 얻으려면 SqlSessionFactory객체가 필요

```
root-context.xml  mybatis-config.xml  board-mapping.xml  BoardDAO.java  SqlSessionFactoryBean.java  BoardServiceClient.java

1 package kr.ac.inje.comsi.board;
2
3 import java.io.Reader;
4
5 import org.apache.ibatis.io.Resources;
6 import org.apache.ibatis.session.SqlSession;
7 import org.apache.ibatis.session.SqlSessionFactory;
8 import org.apache.ibatis.session.SqlSessionFactoryBuilder;
9
10 public class SqlSessionFactoryBean {
11     private static SqlSessionFactory sessionFactory = null;
12
13     static {
14         try {
15             if (sessionFactory == null) {
16                 Reader reader = Resources.getResourceAsReader("mybatis-config.xml");
17                 sessionFactory = new SqlSessionFactoryBuilder().build(reader);
18             }
19         } catch (Exception e) {
20             e.printStackTrace();
21         }
22     }
23
24     public static SqlSession getSqlSessionInstance() {
25         return sessionFactory.openSession();
26     }
27 }
```

→ SqlSession객체를 넘긴다.

# BoardDAO 수정

root-context.xml mybatis-config.xml board-mapping.xml BoardDAO.java SqlSessionFactoryBean.java BoardServiceClient.java

```
1 package kr.ac.inje.comsi.board.impl;
2
3 import java.util.List;
4
5 import org.apache.ibatis.session.SqlSession;
6
7 import kr.ac.inje.comsi.board.BoardVO;
8 import kr.ac.inje.comsi.board.SqlSessionFactoryBean;
9
10 public class BoardDAO {
11
12     private SqlSession mybatis;
13
14     public BoardDAO(){
15         mybatis = SqlSessionFactoryBean.getSqlSessionInstance();
16     }
17
18     public void insertBoard(BoardVO vo) {
19         mybatis.insert("BoardDAO.insertBoard", vo);
20         mybatis.commit();
21     }
22
23
24     public void updateBoard(BoardVO vo) {
25         mybatis.update("BoardDAO.updateBoard", vo);
26         mybatis.commit();
27     }
28 }
```



## 이어서...

```
29 public void deleteBoard(BoardVO vo) {  
30     mybatis.delete("BoardDAO.deleteBoard", vo);  
31     mybatis.commit();  
32 }  
33  
34  
35 public BoardVO getBoard(BoardVO vo) {  
36     return (BoardVO) mybatis.selectOne("BoardDAO.getBoard", vo);  
37 }  
38  
39 public List<BoardVO> getBoardList(BoardVO vo) {  
40     return mybatis.selectList("BoardDAO.getBoardList", vo);  
41 }  
42  
43 }  
44
```

# 테스트 클라이언트-BoardServiceClient 클래스 수정

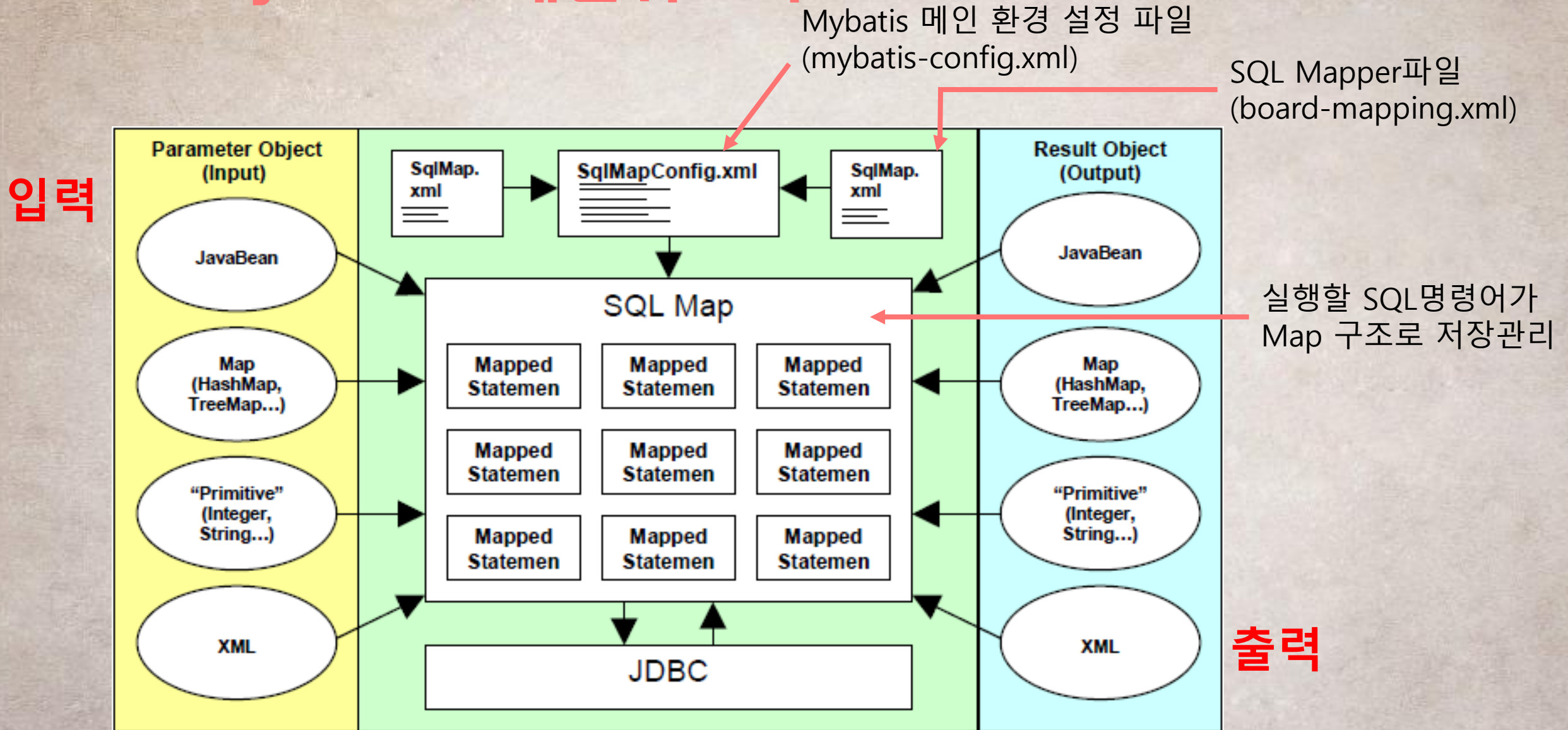
```
root-context.xml  mybatis-config.xml  board-mapping.xml  BoardDAO.java  SqlSessionFactoryBean.java  BoardServiceClient.java  ⌵
1 package kr.ac.inje.comsi.board;
2
3 import java.util.List;
4
5 import kr.ac.inje.comsi.board.impl.BoardDAO;
6
7 public class BoardServiceClient {
8
9     public static void main(String[] args) {
10         BoardDAO boardDAO = new BoardDAO();
11
12         BoardVO vo = new BoardVO();
13         vo.setTitle("myBatis 제목");
14         vo.setWriter("김길동");
15         vo.setContent("myBatis 내용입니다....");
16         //boardDAO.insertBoard(vo);
17
18         List<BoardVO> boardList = boardDAO.getBoardList(vo);
19         for (BoardVO board:boardList) {
20             System.out.println("-->" + board.toString());
21         }
22     }
23
24 }
```

```
Console  ⌵  Progress  Problems  Markers
<terminated> BoardServiceClient [Java Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (2018. 11. 12. 오후 10:16:53)
-->BoardVO [seq=5, title=글 등록, writer=글 등록, content=글 수정테스트, regDate=2018-09-10, cnt=0]
-->BoardVO [seq=4, title=JDBC 테스트2, writer=관리자, content=JDBC 테스트2....., regDate=2018-05-27, cnt=0]
-->BoardVO [seq=3, title=JDBC 테스트, writer=관리자, content=JDBC 테스트....., regDate=2018-05-27, cnt=0]
-->BoardVO [seq=2, title=임시 제목, writer=홍길동, content=임시 내용 ....., regDate=2018-05-12, cnt=0]
-->BoardVO [seq=1, title=가입인사, writer=관리자, content=잘 부탁드립니다...., regDate=2018-04-08, cnt=0]
|
```



## 02. Mapper XML 파일 설정

## 2.1.1 Mybatis 프레임워크 구조





## 2.1.2 SQL Mapper 파일 구조

```
<?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">
```

DTD 선언 부분

```
<mapper namespace="BoardDAO">
```

mapper 루트 엘리먼트

```
<insert id="insertBoard">  
    insert into board(seq, title, writer, content)  
    values((select nvl(max(seq),0)+1 from board),#{title}, #{writer},#{content})  
</insert>
```

```
<select id="getBoard" resultType="board">  
    select * from board where seq=#{seq}  
</select>
```

```
<select id="getBoardList" resultType="board">  
    select * from board order by seq desc  
</select>
```

```
<delete id="deleteBoard">  
    delete board where seq=#{seq}  
</delete>
```

```
<update id="updateBoard">  
    update board set title=#{title}, content=#{content} where seq=#{seq}  
</update>
```

SQL 매핑 부분

```
</mapper>
```

# Mapper 파일과 DAO 클래스 연결

Mapper XML	<pre>&lt;mapper namespace="BoardDAO"&gt;      &lt;insert id="insertBoard"&gt;         insert into board(seq, title, writer, content)         values((select nvl(max(seq),0)+1 from board),#{title}, #{writer},#{content})     &lt;/insert&gt;  &lt;/mapper&gt;</pre>
DAO 클래스	<pre>public void insertBoard(BoardVO vo) {     mybatis.insert("BoardDAO.insertBoard", vo);     mybatis.commit(); }</pre>

- Mapper 파일에 SQL 명령어를 등록할 때는 SQL 구문의 종류에 따라 적절한 엘리먼트를 사용한다.
- INSERT 구문은 <insert>, SELECT 구문은 <select> 엘리먼트를 사용
- 각 엘리먼트에서 사용할 수 있는 속성들이 다르므로 그 용도와 의미를 이해해야 한다.



## 2.1.3 <select> 엘리먼트

Mapper XML	<pre>&lt;select id="getBoard" parameterType="board" resultType="board"&gt;     select * from board where seq=#{seq} &lt;/select&gt;  &lt;select id="getBoardList" resultType="board"&gt;     select * from board order by seq desc &lt;/select&gt;</pre>
------------	--

- id – 필수 속성으로 Mapper 파일들 내에서 유일해야하며 DAO에서 호출가능하다.
- parameterType – SQL 실행에 필요한 데이터를 외부로부터 받아야 할 때 사용하며 일반적으로 기본형이나 VO형태의 클래스를 지정
- resultType – SQL 구문이 실행된 ResultSet 결과를 매핑할 객체 지정



## 2.1.4 기타 엘리먼트

<p>&lt;insert&gt; 엘리먼트</p>	<pre>&lt;insert id="insertBoard" parameterType="board"&gt;     insert into board(seq, title, writer, content)         values((select nvl(max(seq),0)+1 from board),#{title}, #{writer},#{content}) &lt;/insert&gt;</pre>
<p>&lt;update&gt; 엘리먼트</p>	<pre>&lt;update id="updateBoard" parameterType="board"&gt;     update board set title=#{title}, content=#{content} where seq=#{seq} &lt;/update&gt;</pre>
<p>&lt;delete&gt; 엘리먼트</p>	<pre>&lt;delete id="deleteBoard" parameterType="board"&gt;     delete board where seq=#{seq} &lt;/delete&gt;</pre>

## 2.2 SQL Mapper XML 추가 설정

- 2.2.1 resultMap 속성 설정
  - 일반적으로 resultType 속성으로 SQL 실행결과를 넘겨주는 방식을 사용한다.
    - 하나의 테이블 결과를 넘겨줄 때 테이블의 VO객체 사용
  - 2개 테이블을 join하여 넘겨줄 때는 적용할 수 없다. → resultMap 사용
- 2.2.2 CDATA Section 사용

```
<select id="getBoard" parameterType="board" resultType="board">
    select * from board where seq=#{seq}
</select>
```

- CDATA(Character DATA)는 XML 파서에 의해서 해석되지 않는다. 즉, CDATA 안의 내용이 그대로 DB에 전달되어 실행된다.

```
<select id="getBoard" parameterType="board" resultType="board">
    <![CDATA[
        select * from board where seq=#{seq} < 5
    ]]>
</select>
```



## 2.3 Mybatis JAVA API

### • 2.3.1 SqlSessionFactoryBuilder 클래스

- Mybatis로 DAO 클래스의 CRUD메소드를 구현하려면 Mybatis에서 제공하는 SqlSession객체를 사용해야 함.
- SqlSession객체는 SqlSessionFactory로부터 얻어야한다.
- 따라서, 가장 먼저 할 작업은 SqlSessionFactory 객체를 생성하는 일임.
- SqlSessionFactory는 SqlSessionFactoryBuild의 build()메소드를 이용
  - Mybatis 설정 파일()을 로딩하여 SqlSessionFactory객체를 생성

```
Reader reader = Resources.getResourceAsReader("mybatis-config.xml");  
sessionFactory = new SqlSessionFactoryBuilder().build(reader);
```

### • 2.3.2 SqlSessionFactory 클래스

- SqlSession 객체에 대한 생성 역할을 수행
- SqlSessionFactory객체는 openSession()이라는 메소드를 통해 SqlSession 객체를 획득

```
sessionFactory.openSession();
```



## • 2.3.3 유틸리티 클래스 작성 - 싱글톤 디자인 패턴으로 작성

```
root-context.xml  mybatis-config.xml  board-mapping.xml  BoardDAO.java  SqlSessionFactoryBean.java  BoardServiceClient.java

1 package kr.ac.inje.comsi.board;
2
3 import java.io.Reader;
4
5 import org.apache.ibatis.io.Resources;
6 import org.apache.ibatis.session.SqlSession;
7 import org.apache.ibatis.session.SqlSessionFactory;
8 import org.apache.ibatis.session.SqlSessionFactoryBuilder;
9
10 public class SqlSessionFactoryBean {
11     private static SqlSessionFactory sessionFactory = null;
12
13     static {
14         try {
15             if (sessionFactory == null) {
16                 Reader reader = Resources.getResourceAsReader("mybatis-config.xml");
17                 sessionFactory = new SqlSessionFactoryBuilder().build(reader);
18             }
19         } catch (Exception e) {
20             e.printStackTrace();
21         }
22     }
23
24     public static SqlSession getSqlSessionInstance() {
25         return sessionFactory.openSession();
26     }
27 }
```

## 2.3.4 SqlSession 객체

- (1) selectOne() : 오직 하나의 데이터를 검색하는 SQL구문 실행할 때
  - public Object selectOne(String statement)
  - public Object selectOne(String statement, Object parameter)
  - statement 매개변수는 Mapper XML 파일에 등록된 SQL의 아이디(ID)
- (2) selectList() : 여러 개의 데이터가 검색되는 SQL구문을 실행할 때
  - public Object selectList(String statement)
  - public Object selectList(String statement, Object parameter)
- (3) insert(), update(), delete() : 각각 INSERT, UPDATE, DELETE SQL구문을 실행할 때
  - public int insert(String statement, Object parameter)
  - public int update(String statement, Object parameter) throws SQLException
  - public int delete(String statement, Object parameter) throws SQLException

➔ 사용 방법은 BoardDAO 클래스에서 확인



## 03. 스프링과 Mybatis 연동



## 3.2 Mybatis 설정 파일 복사 및 설정

```
root-context.xml | mybatis-config.xml | board-mapping.xml | BoardDAO.java | SqlSessionFactoryBean.java | BoardServiceClient.java
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
3     "http://mybatis.org/dtd/mybatis-3-config.dtd">
4
5 <configuration>
6     <!-- <properties resource="db.properties"/> -->
7     <typeAliases>
8         <typeAlias alias="board" type="kr.ac.inje.comsi.board.BoardVO"/>
9     </typeAliases>
10    <!-- <environments default="development">
11        <environment id="development">
12            <transactionManager type="JDBC"/>
13            <dataSource type="POOLED">
14                <property name="driver" value="org.h2.Driver"></property>
15                <property name="url" value="jdbc:h2:D:/workspace/h2db/db"></property>
16                <property name="username" value="sa"></property>
17                <property name="password" value=""></property>
18            </dataSource>
19        </environment>
20    </environments> -->
21
22    <mappers>
23        <mapper resource="mapper/board-mapping.xml"/>
24    </mappers>
25 </configuration>
26
```

주석 처리

## 3.3 스프링 연동: root-context.xml 파일 수정

SqlSessionFactoryBean 클래스를 Bean으로 등록해서 사용

```
root-context... mybatis-conf... BoardDAO.java BoardServic... BoardDAOMyb... applicationC... BoardServic... UserService... »2
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:p="http://www.springframework.org/schema/p"
5     xmlns:context="http://www.springframework.org/schema/context"
6     xmlns:aop="http://www.springframework.org/schema/aop"
7     xmlns:tx="http://www.springframework.org/schema/tx"
8     xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
9         http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd
10        http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.3.xsd
11        http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.3.xsd">
12
13     <context:component-scan base-package="kr.ac.inje.comsi"></context:component-scan>
14
15     <bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
16         <property name="driverClassName" value="org.h2.Driver"></property>
17         <property name="url" value="jdbc:h2:D:/workspace/h2db/db"></property>
18         <property name="username" value="sa"></property>
19         <property name="password" value=""></property>
20     </bean>
21
22     <!-- mybatis-config.xml 파일이 스프링 시작할 때, 같이 동작하도록 설정 -->
23     <bean id="sessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
24         <property name="dataSource" ref="dataSource"></property>
25         <property name="configLocation" value="classpath:mybatis-config.xml"></property>
26     </bean>
27 </beans>
28
```

DataSource 설정

SqlSessionFactoryBean 등록

## 3.4 DAO 클래스 구현-방법1

- SqlSessionDaoSupport 클래스 상속하여 구현

```
SqlSessionFactoryBean.java BoardServiceClient.java BoardDAOMybatis.java applicationContext.xml BoardServiceImpl.java
1 package kr.ac.inje.comsi.board;
2
3 import java.util.List;
4
5 import org.apache.ibatis.session.SqlSessionFactory;
6 import org.mybatis.spring.support.SqlSessionDaoSupport;
7 import org.springframework.beans.factory.annotation.Autowired;
8 import org.springframework.stereotype.Repository;
9
10 @Repository
11 public class BoardDAOMybatis extends SqlSessionDaoSupport {
12
13     @Autowired // 스프링 설정 파일의 SqlSessionFactoryBean을 인자로 받아 설정
14     public void setSqlSessionFactory(SqlSessionFactory sqlSessionFactory) {
15         super.setSqlSessionFactory(sqlSessionFactory);
16     }
17
18     public void insertBoard(BoardVO vo) {
19         System.out.println("==>Mybatis로 insertBoard기능 처리");
20         getSqlSession().insert("BoardDAO.insertBoard", vo);
21     }
22
23
24     public void updateBoard(BoardVO vo) {
25         System.out.println("==>Mybatis로 updateBoard기능 처리");
26         getSqlSession().update("BoardDAO.updateBoard", vo);
27     }
28 }
```



## 이어서...

```
28
29 public void deleteBoard(BoardVO vo) {
30     System.out.println("=>Mybatis로 deleteBoard기능 처리");
31     getSession().delete("BoardDAO.deleteBoard", vo);
32 }
33
34
35 public BoardVO getBoard(BoardVO vo) {
36     System.out.println("=>Mybatis로 getBoard기능 처리");
37     return (BoardVO) getSession().selectOne("BoardDAO.getBoard", vo);
38 }
39
40 public List<BoardVO> getBoardList(BoardVO vo) {
41     System.out.println("=>Mybatis로 getBoardList기능 처리");
42     return getSession().selectList("BoardDAO.getBoardList", vo);
43 }
44
45 }
46
```

# BoardServiceImpl 클래스 수정

```
BoardServiceClient.java BoardDAOMybatis.java applicationContext.xml BoardServiceImpl.java
1 package kr.ac.inje.comsi.board.impl;
2
3 import java.util.List;
4
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Service;
7
8 import kr.ac.inje.comsi.board.BoardDAOMybatis;
9 import kr.ac.inje.comsi.board.BoardService;
10 import kr.ac.inje.comsi.board.BoardVO;
11
12 @Service("boardService")
13 public class BoardServiceImpl implements BoardService {
14
15     @Autowired
16     private BoardDAOMybatis boardDAOMybatis;
17     //private BoardDAO boardDAO;
18
19
20     @Override
21     public void insertBoard(BoardVO vo) {
22         //boardDAO.insertBoard(vo);
23         boardDAOMybatis.insertBoard(vo);
24     }
25 }
```



# 이어서...

```
26 @Override
27 public void updateBoard(BoardVO vo) {
28     //boardDAO.updateBoard(vo);
29     boardDAOMybatis.updateBoard(vo);
30 }
31
32 @Override
33 public void deleteBoard(BoardVO vo) {
34     //boardDAO.deleteBoard(vo);
35     boardDAOMybatis.deleteBoard(vo);
36 }
37
38 @Override
39 public BoardVO getBoard(BoardVO vo) {
40     //return boardDAO.getBoard(vo);
41     return boardDAOMybatis.getBoard(vo);
42 }
43
44 @Override
45 public List<BoardVO> getBoardList(BoardVO vo) {
46     //return boardDAO.getBoardList(vo);
47     return boardDAOMybatis.getBoardList(vo);
48 }
49 }
50
```

# BoardServiceClient 클래스...

BoardServiceClient.java BoardDAOMybatis.java applicationContext.xml BoardServiceImpl.java

```
1 package kr.ac.inje.comsi.board;
2
3 import java.util.List;
4
5 import org.springframework.context.support.AbstractApplicationContext;
6 import org.springframework.context.support.GenericXmlApplicationContext;
7
8 public class BoardServiceClient {
9
10     public static void main(String[] args) {
11         // 1. Spring 컨테이너를 구동
12         AbstractApplicationContext container = new GenericXmlApplicationContext("applicationContext.xml");
13
14         // 2. Spring 컨테이너로부터 UserServiceImpl 객체를 Lookup
15         BoardService boardService = (BoardService) container.getBean("boardService");
16
17         BoardVO vo = new BoardVO();
18         vo.setTitle("myBatis 제목");
19         vo.setWriter("김길동");
20         vo.setContent("myBatis 내용입니다....");
21         //boardDAO.insertBoard(vo);
22
23         List<BoardVO> boardList = boardService.getBoardList(vo);
24         for (BoardVO board:boardList) {
25             System.out.println("-->" + board.toString());
26         }
27         container.close();
28     }
29
30 }
```



# 실행 결과...

```
Console Progress Problems Markers
<terminated> BoardServiceClient [Java Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (2018. 11. 15. 오후 9:44:36)
INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions from class path resource [applicationContext.xml]
INFO : org.springframework.context.support.GenericXmlApplicationContext - Refreshing org.springframework.context.support.GenericXmlApplicationContext
INFO : org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor - JSR-330 'javax.inject.Inject' annotation found and supported
==>Mybatis로 getBoardList기능 처리
-->BoardVO [seq=5, title=글 등록, writer=글 등록, content=글 수정테스트, regDate=2018-09-10, cnt=0]
-->BoardVO [seq=4, title=JDBC 테스트2, writer=관리자, content=JDBC 테스트2....., regDate=2018-05-27, cnt=0]
-->BoardVO [seq=3, title=JDBC 테스트, writer=관리자, content=JDBC 테스트....., regDate=2018-05-27, cnt=0]
-->BoardVO [seq=2, title=임시 제목, writer=홍길동, content=임시 내용 ....., regDate=2018-05-12, cnt=0]
-->BoardVO [seq=1, title=가입인사, writer=관리자, content=잘 부탁드립니다...., regDate=2018-04-08, cnt=0]
```

## 3.4 DAO 클래스 구현-방법2

- SqlSessionTemplate클래스를 Bean으로 등록하여 사용

```
BoardServiceClient.java BoardDAOMybatis.java applicationContext.xml BoardServiceImpl.java
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:p="http://www.springframework.org/schema/p"
5     xmlns:context="http://www.springframework.org/schema/context"
6     xmlns:aop="http://www.springframework.org/schema/aop"
7     xmlns:tx="http://www.springframework.org/schema/tx"
8     xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd
9         http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd
10        http://www.springframework.org/schema/aop http://www.springframework.org/schema/aop/spring-aop-4.3.xsd
11        http://www.springframework.org/schema/tx http://www.springframework.org/schema/tx/spring-tx-4.3.xsd">
12
13     <context:component-scan base-package="kr.ac.inje.comsi"></context:component-scan>
14
15     <bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
16         <property name="driverClassName" value="org.h2.Driver"></property>
17         <property name="url" value="jdbc:h2:D:/workspace/h2db/db"></property>
18         <property name="username" value="sa"></property>
19         <property name="password" value=""></property>
20     </bean>
21
22     <!-- mybatis-config.xml 파일이 스프링 시작할 때, 같이 동작하도록 설정 -->
23     <bean id="sqlSession" class="org.mybatis.spring.SqlSessionFactoryBean">
24         <property name="dataSource" ref="dataSource"></property>
25         <property name="configLocation" value="classpath:mybatis-config.xml"></property>
26     </bean>
27     <bean class="org.mybatis.spring.SqlSessionTemplate">
28         <!-- sqlSessionTemplate클래스는 setter 메소드가 없어서 생성자로 주입함 -->
29         <constructor-arg ref="sqlSession"></constructor-arg>
30     </bean>
31 </beans>
```



# BoardDAOMybatis 클래스 수정

BoardServiceClient.java BoardDAOMybatis.java applicationContext.xml BoardServiceImpl.java

```
1 package kr.ac.inje.comsi.board;
2
3 import java.util.List;
4
5 import org.mybatis.spring.SqlSessionTemplate;
6 import org.springframework.beans.factory.annotation.Autowired;
7 import org.springframework.stereotype.Repository;
8
9 @Repository
10 public class BoardDAOMybatis {
11
12     @Autowired
13     private SqlSessionTemplate mybatis;
14
15     public void insertBoard(BoardVO vo) {
16         System.out.println("==>Mybatis로 insertBoard기능 처리");
17         mybatis.insert("BoardDAO.insertBoard", vo);
18     }
19
20
21     public void updateBoard(BoardVO vo) {
22         System.out.println("==>Mybatis로 updateBoard기능 처리");
23         mybatis.update("BoardDAO.updateBoard", vo);
24     }
25 }
```

## 이어서...

```
26 public void deleteBoard(BoardVO vo) {  
27     System.out.println("==>Mybatis로 deleteBoard기능 처리");  
28     mybatis.delete("BoardDAO.deleteBoard", vo);  
29 }  
30  
31  
32 public BoardVO getBoard(BoardVO vo) {  
33     System.out.println("==>Mybatis로 getBoard기능 처리");  
34     return (BoardVO) mybatis.selectOne("BoardDAO.getBoard", vo);  
35 }  
36  
37 public List<BoardVO> getBoardList(BoardVO vo) {  
38     System.out.println("==>Mybatis로 getBoardList기능 처리");  
39     return mybatis.selectList("BoardDAO.getBoardList", vo);  
40 }  
41  
42 }  
43  
44
```



# 실행 결과

```
Console Progress Problems Markers
<terminated> BoardServiceClient [Java Application] C:\Program Files\Java\jre1.8.0_171\bin\javaw.exe (2018. 11. 15. 오후 9:58:04)
INFO : org.springframework.beans.factory.xml.XmlBeanDefinitionReader - Loading XML bean definitions from class path resource [applicationContext.xml]
INFO : org.springframework.context.support.GenericXmlApplicationContext - Refreshing org.springframework.context.support.GenericXmlApplicationContext
INFO : org.springframework.beans.factory.annotation.AutowiredAnnotationBeanPostProcessor - JSR-330 'javax.inject.Inject' annotation found and processed
==>Mybatis로 getBoardList기능 처리
-->BoardVO [seq=5, title=글 등록, writer=글 등록, content=글 수정테스트, regDate=2018-09-10, cnt=0]
-->BoardVO [seq=4, title=JDBD 테스트2, writer=관리자, content=JDBC 테스트2....., regDate=2018-05-27, cnt=0]
-->BoardVO [seq=3, title=JDBD 테스트, writer=관리자, content=JDBC 테스트....., regDate=2018-05-27, cnt=0]
-->BoardVO [seq=2, title=임시 제목, writer=홍길동, content=임시 내용 ....., regDate=2018-05-12, cnt=0]
-->BoardVO [seq=1, title=가입인사, writer=관리자, content=잘 부탁드립니다...., regDate=2018-04-08, cnt=0]
INFO : org.springframework.context.support.GenericXmlApplicationContext - Closing org.springframework.context.support.GenericXmlApplicationContext
```

**WEB에서 보기.**



# root-context.xml파일 수정(applicationContext.xml과 동일)

```
BoardServiceClient.java BoardDAOMybatis.java applicationContext.xml BoardServiceImpl.java root-context.xml
5  xmlns:mvc="http://www.springframework.org/schema/mvc"
6  xmlns:mybatis-spring="http://mybatis.org/schema/mybatis-spring"
7  xmlns:jdbc="http://www.springframework.org/schema/jdbc"
8  xsi:schemaLocation="http://www.springframework.org/schema/jdbc http://www.springframework.org/schema/jdbc/spring-jdbc-4.3.xsd
9  http://www.springframework.org/schema/mvc http://www.springframework.org/schema/mvc/spring-mvc-4.3.xsd
10 http://mybatis.org/schema/mybatis-spring http://mybatis.org/schema/mybatis-spring-1.2.xsd
11 http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans-4.3.xsd
12 http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd">
13
14 <context:component-scan base-package="kr.ac.inje.comsi.board"></context:component-scan>
15
16 <bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSource">
17     <property name="driverClassName" value="org.h2.Driver"></property>
18     <property name="url" value="jdbc:h2:D:/workspace/h2db/db"></property>
19     <property name="username" value="sa"></property>
20     <property name="password" value=""></property>
21 </bean>
22
23 <!-- mybatis-config.xml 파일이 스프링 시작할 때, 같이 동작하도록 설정 -->
24 <bean id="sqlSession" class="org.mybatis.spring.SqlSessionFactoryBean">
25     <property name="dataSource" ref="dataSource"></property>
26     <property name="configLocation" value="classpath:mybatis-config.xml"></property>
27 </bean>
28 <bean class="org.mybatis.spring.SqlSessionTemplate">
29     <!-- SqlSessionTemplate클래스는 setter 메소드가 없어서 생성자로 주입함 -->
30     <constructor-arg ref="sqlSession"></constructor-arg>
31 </bean>
32
33 </beans>
34
```



## 글 목록

환영합니다. [Logout](#)

제목 ▼ <input type="text"/> 검색				
번호	제목	작성자	등록일	조회수
5	<a href="#">글 등록</a>	글 등록	2018-09-10	0
4	<a href="#">JDBD 테스트2</a>	관리자	2018-05-27	0
3	<a href="#">JDBD 테스트</a>	관리자	2018-05-27	0
2	<a href="#">임시 제목</a>	홍길동	2018-05-12	0
1	<a href="#">가입인사</a>	관리자	2018-04-08	0

[새글 등록](#)

[Log-out](#)

제목	글 등록
작성자	글 등록
내용	글 수정테스트
등록일	2018-09-10
조회수	0
글 수정	

[글등록](#) [글삭제](#) [글목록](#)