



박 경 태

comsi.java@gmail.com

고급 자바 프로그래밍
: STS를 이용한 Spring 프로그래밍

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

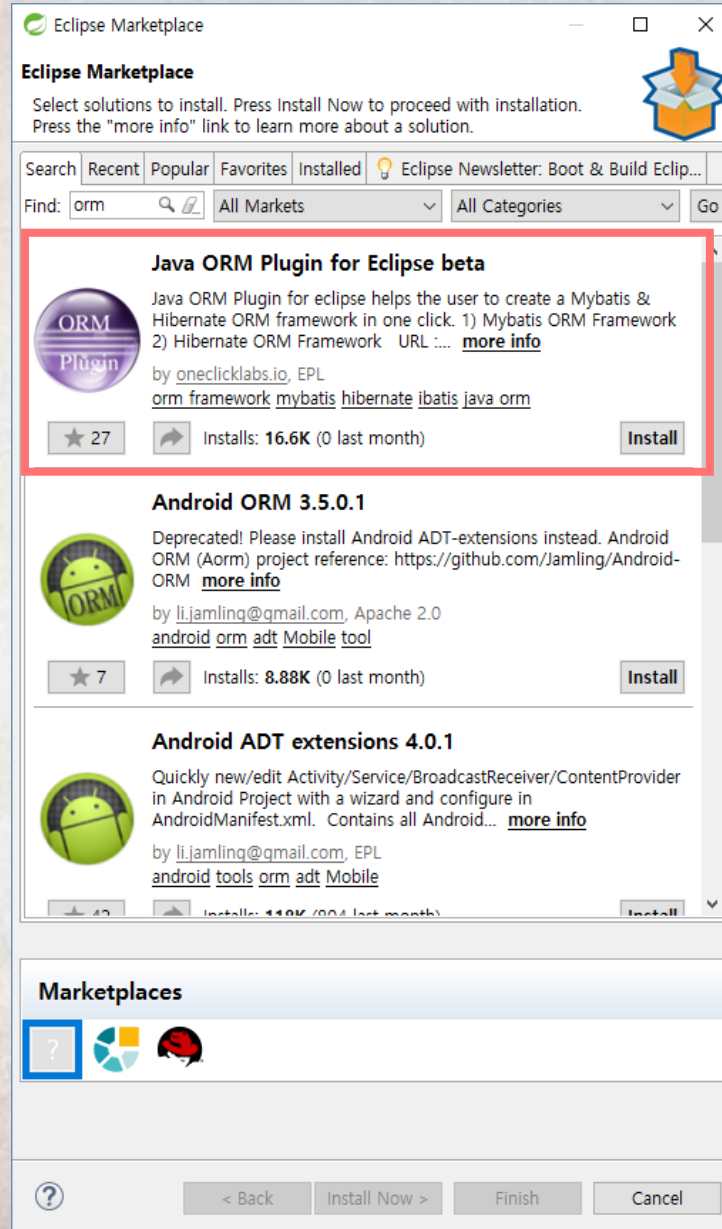
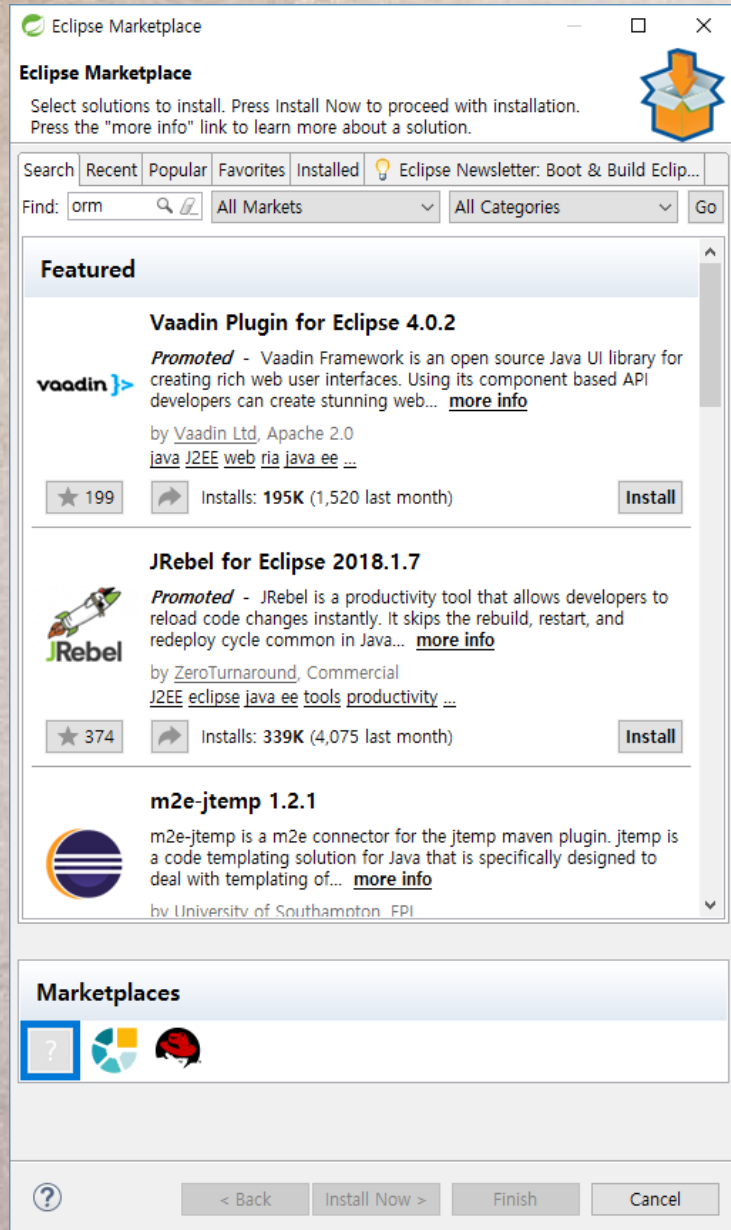
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
  "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<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 π π



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, and 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

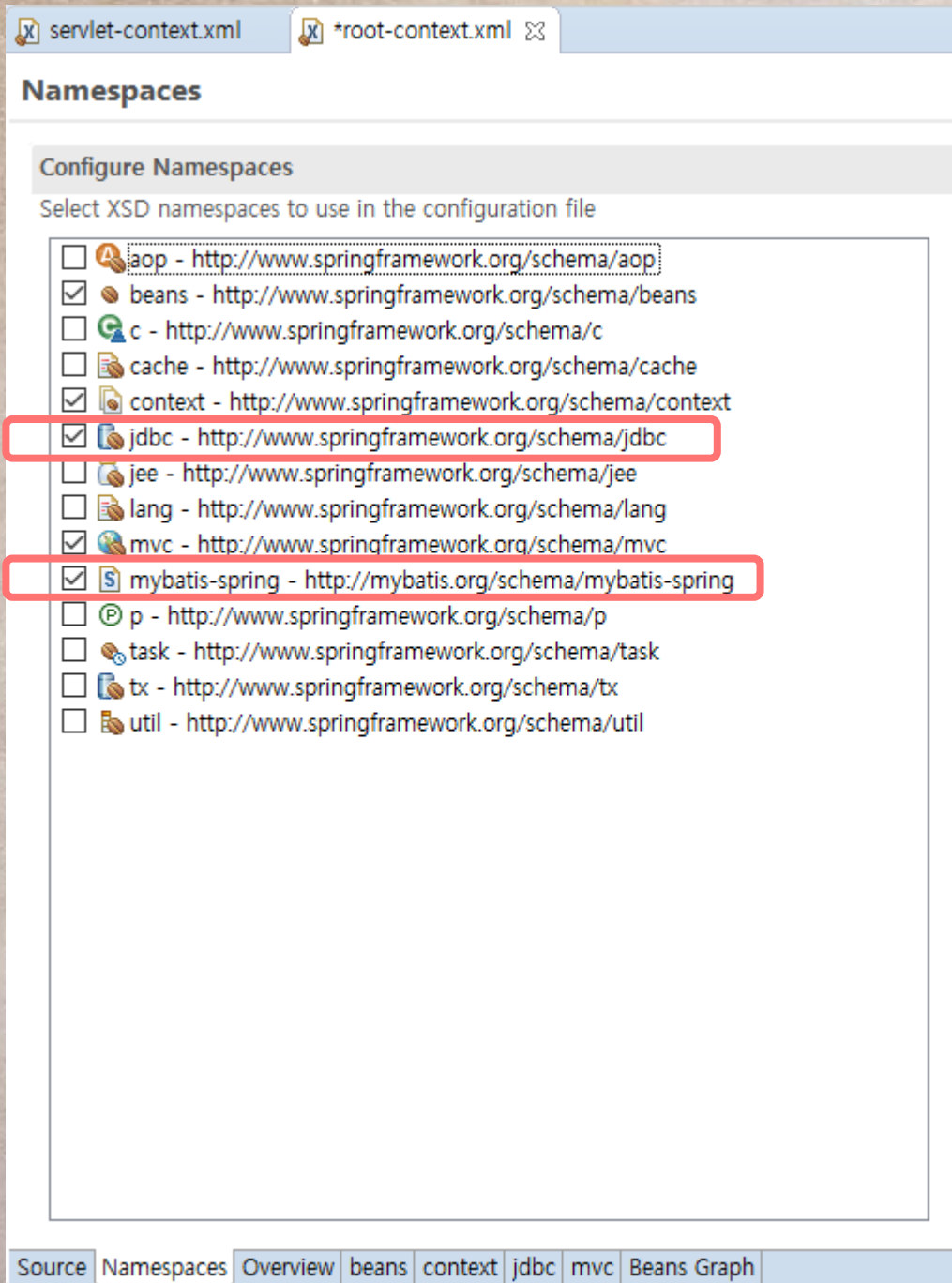
> src/test/resources

▼ 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

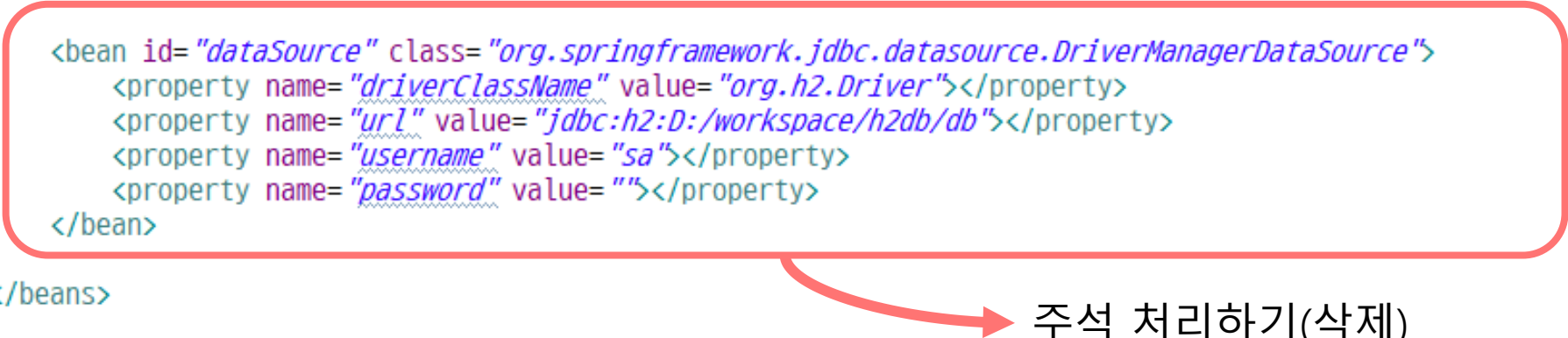


root-context.xml 파일

- Namespace 항목 추가하기
 - jdbc
 - mybatis-spring

1.4 root-context.xml 수정-DataSource 추석 추가

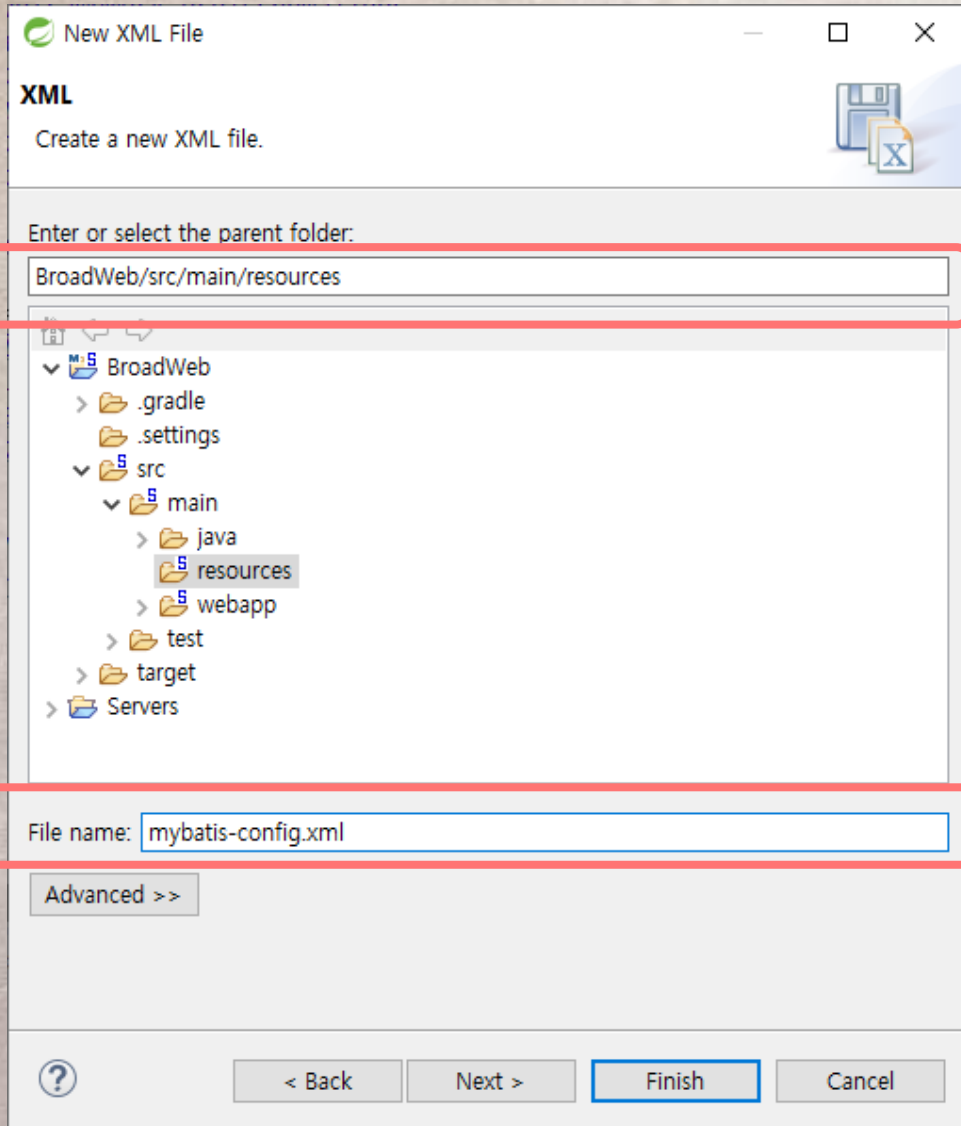
```
servlet-context.xml  root-context.xml  applicationContext.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4     xmlns:context="http://www.springframework.org/schema/context"
5     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

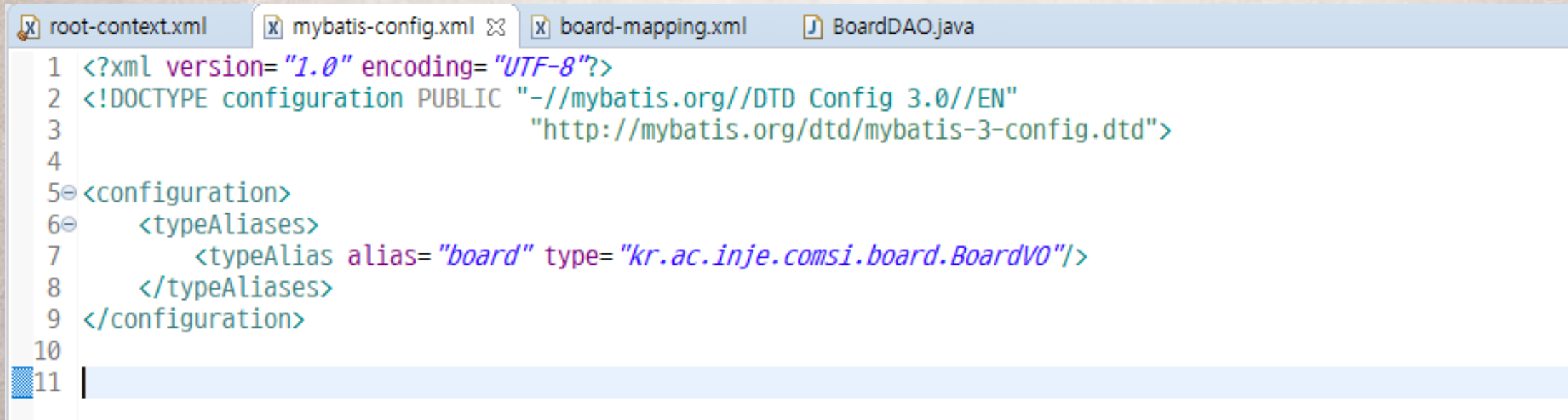


BroadWeb/src/main/resources

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 |
```

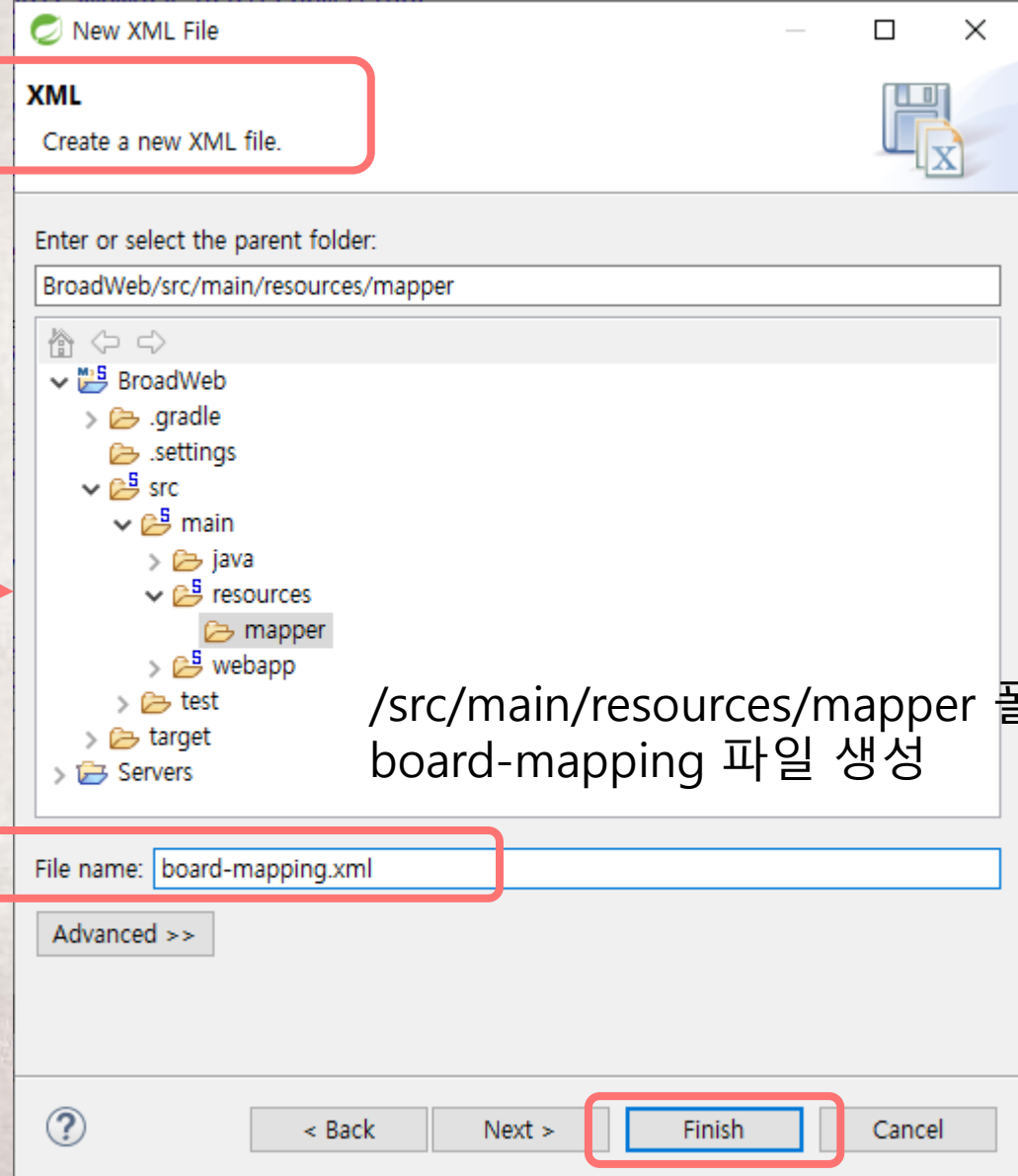
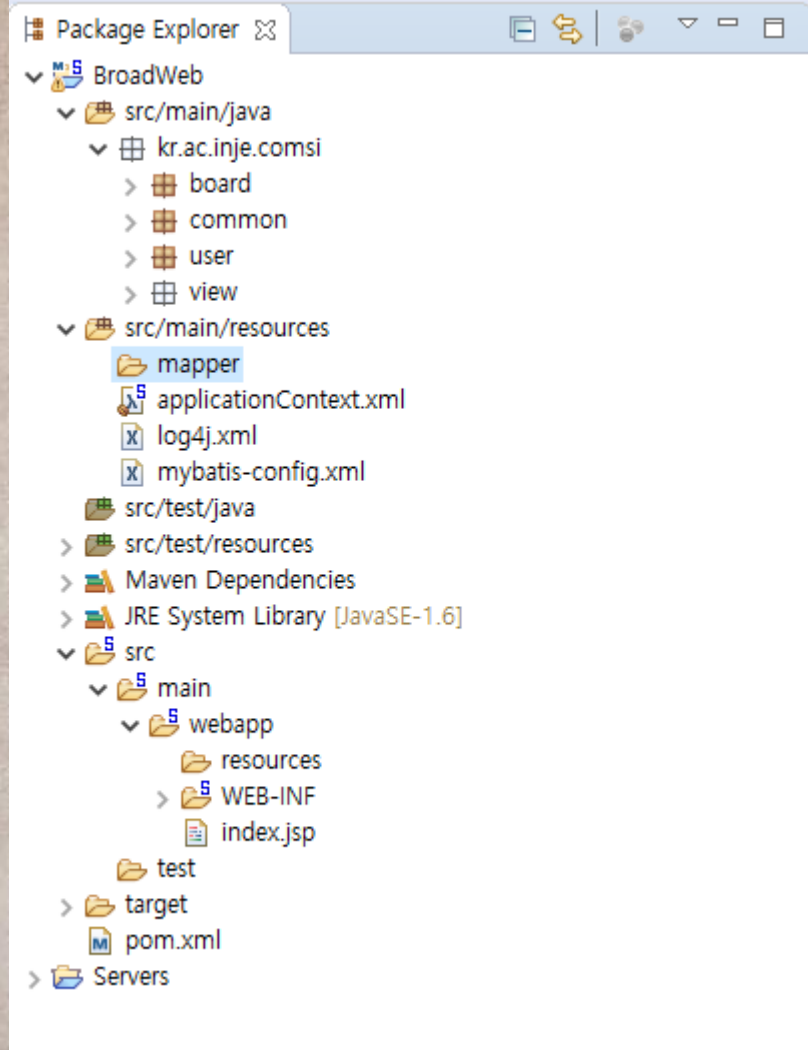
mybatis-config.xml 내용 파일 추가

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

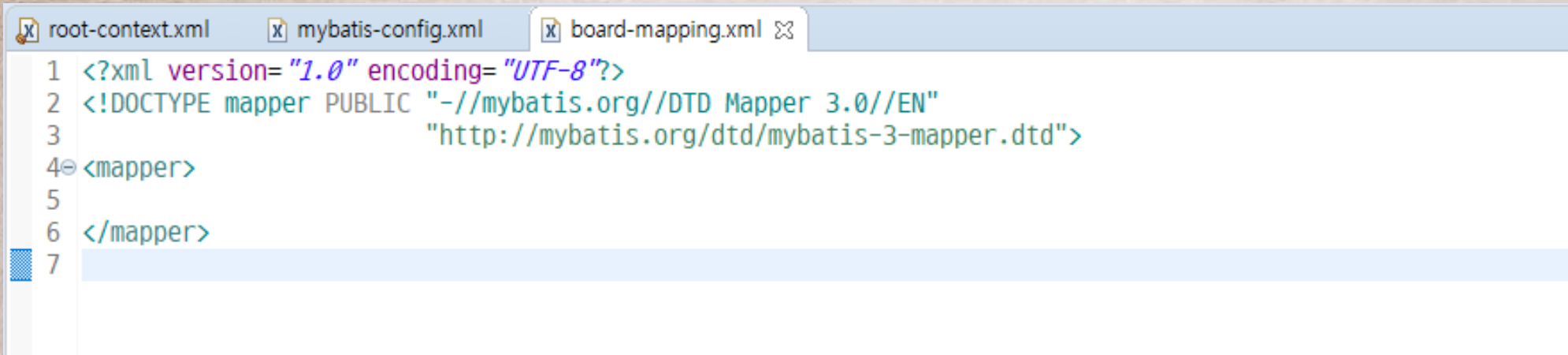
1  <?xml version="1.0" encoding="UTF-8"?>
2  <!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
3      "http://mybatis.org/dtd/mybatis-3-config.dtd">
4
5  <configuration>
6      <!-- <properties resource="db.properties"/> -->
7      <typeAliases>
8          <typeAlias alias="board" type="kr.ac.inje.comsi.board.BoardVO"/>
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"/>  sql mapper 파일 위치 설정
24     </mappers>
25 </configuration>
26
```

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

1.6 SQL Mapper 파일 설정 – Board Table



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

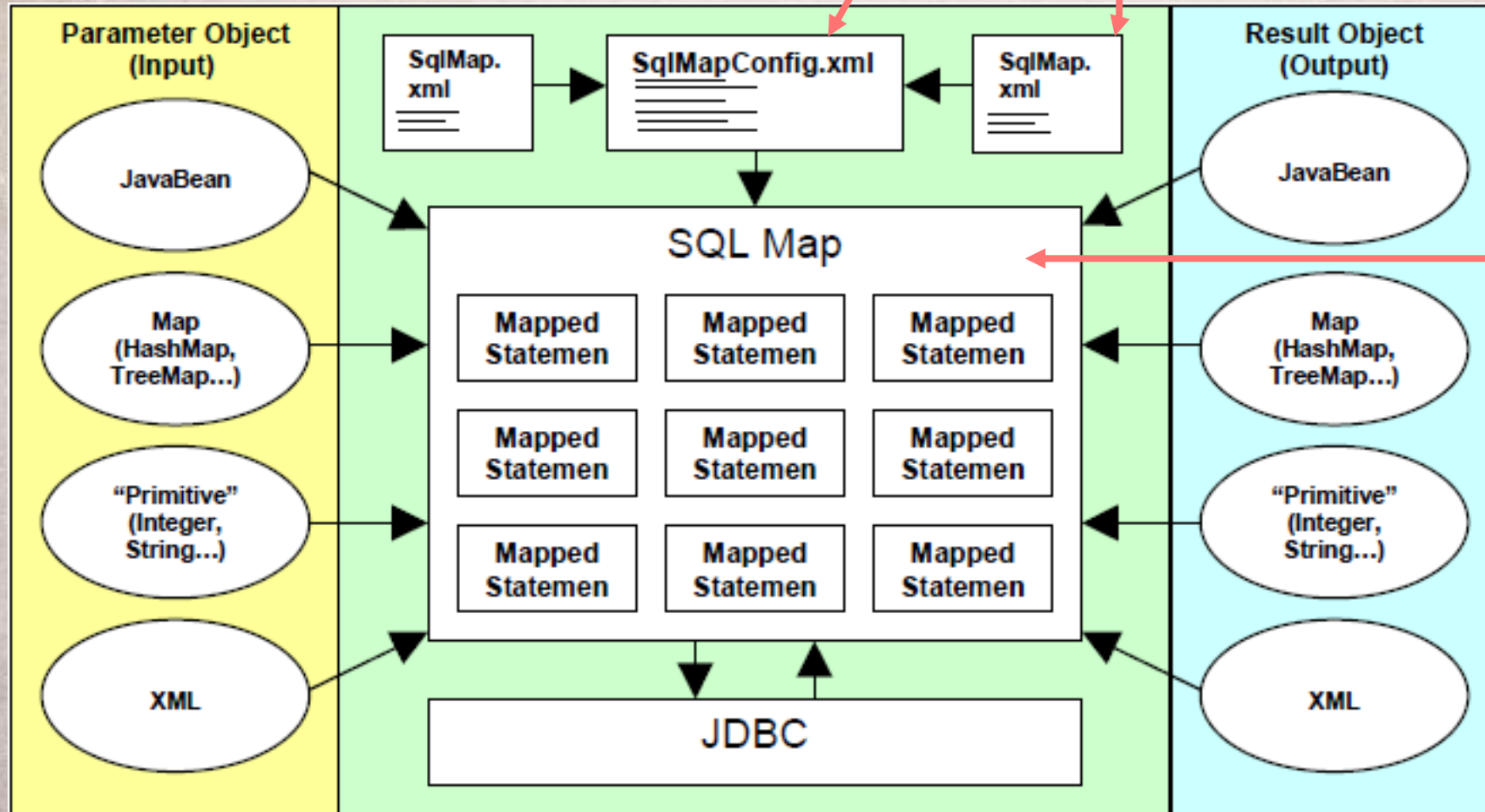
```
root-context.xml  mybatis-config.xml  board-mapping.xml  BoardDAO.java  SqlSessionFactoryBean.java  BoardServiceClient.java  ⌵
1 package kr.ac.inje.comsi.board;
2
3 import java.util.List;
4
5 import kr.ac.inje.comsi.board.impl.BoardDAO;
6
7 public class BoardServiceClient {
8
9     public static 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]
|
```


01. Mapper XML 파일 설정

Mybatis 프레임워크 구조

입력



Mybatis 메인 환경 설정 파일
(mybatis-config.xml)

SQL Mapper파일
(board-mapping.xml)

실행할 SQL 명령어가
Map 구조로 저장관리

출력

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">
```

```
<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>
```