



## Day12; 20220920

📅 날짜	
📁 유형	
🏷 태그	

### arithmeticError.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>기본 예러 화면(ArithmeticExpt)</title>
</head>
<body bgcolor="#ffffff" text="000000">

    <table width="100%" border="1" cellspacing="0" cellpadding="0">
        <tr>
            <td align="center" bgcolor="orange"><b>ArithmeticExpt 예러 화면입니다.</b></td>
        </tr>
    </table>

    <table width="100%" border="1" cellspacing="0" cellpadding="0" align="center">
        <tr>
            <td align="center">
                <br><br><br><br>
                Message : ${ exception.message }
                <br><br><br><br>
            </td>
        </tr>
    </table>

</body>
</html>
```

### CommonExceptionHandler.java

```
package com.springproj.biz.common;

import org.springframework.web.bind.annotation.ControllerAdvice;
import org.springframework.web.bind.annotation.ExceptionHandler;
import org.springframework.web.servlet.ModelAndView;
```

```

@ControllerAdvice("com.springproj.controller") // 예외처리 관련 Advice
public class CommonExceptionHandler {

    @ExceptionHandler(ArithmeticException.class) // controller 패키지에서 ArithmeticException에러가 났을 때 불러준다.(.class)
    public ModelAndView handleArithmeticException(Exception e) {
        ModelAndView mav = new ModelAndView();

        mav.addObject("exception", e); // exception키값으로 보관을 하겠다.
        mav.setViewName("/common/arithmeticError");

        return mav;
    }

    @ExceptionHandler(NullPointerException.class) // controller 패키지에서 ArithmeticException에러가 났을 때 불러준다.(.class)
    public ModelAndView handleNullPointerException(Exception e) {
        ModelAndView mav = new ModelAndView();

        mav.addObject("exception", e); // exception키값으로 보관을 하겠다.
        mav.setViewName("/common/NullPointerException");

        return mav;
    }
}

```

```

<!-- xml 기반의 예외 처리 -->
<beans:bean id="exceptionResolver" class="org.springframework.web.servlet.handler.SimpleMappingExceptionResolver">
    <beans:property name="exceptionMappings">
        <beans:props>
            <beans:prop key="java.lang.ArithmeticException">
                common/arithmeticError
            </beans:prop>
            <beans:prop key="java.lang.NullPointerException">
                common/nullPointerException
            </beans:prop>
        </beans:props>
    </beans:property>
    <beans:property name="defaultErrorView" value="common/error"/>
</beans:bean>

```

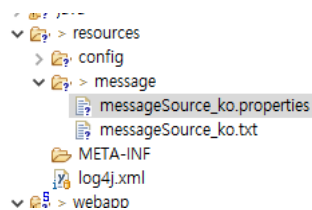
```

<!-- xml 기반의 예외 처리 -->
<beans:bean id="exceptionResolver" class="org.springframework.web.servlet.handler.SimpleMappingExceptionResolver">
    <beans:property name="exceptionMappings">
        <beans:props>
            <beans:prop key="java.lang.ArithmeticException">
                common/arithmeticError
            </beans:prop>
            <beans:prop key="java.lang.NullPointerException">
                common/nullPointerException
            </beans:prop>
        </beans:props>
    </beans:property>
    <beans:property name="defaultErrorView" value="common/error"/>
</beans:bean>

```

```
messageSource_ko.txt × index.jsp messageSource_ko.properties
1# login.jsp
2message.user.login.title=로그인
3message.user.login.id=아이디
4message.user.login.password=비밀번호
5message.user.login.loginBtn=로그인
6
7message.user.login.language.en=영어
8message.user.login.language.ko=한글
9
10# getBoardList.jsp
11message.board.list.mainTitle=게시글 목록
12message.board.list.welcomeMsg=님! 게시판에 오신걸 환영합니다.
13message.board.list.search.condition.title=제목
14message.board.list.search.condition.content=내용
15message.board.list.search.condition.btn=검색
16message.board.list.table.head.seq=번호
17message.board.list.table.head.title=제목
18message.board.list.table.head.writer=작성자
19message.board.list.table.head.regDate=등록일
20message.board.list.table.head.cnt=조회수
21message.board.list.link.insertBoard=새글 등록
```

messageSource\_ko.properties File을 만들고 붙여넣기 하면 자동으로 유니코드로 바뀐다.



```
messageSource_ko.txt × index.jsp messageSource_ko.properties ×
1# login.jsp
2message.user.login.title=\uB85C\uADF8\uC778
3message.user.login.id=\uC544\uC774\uB514
4message.user.login.password=\uBE44\uBC00\uBC88\uD638
5message.user.login.loginBtn=\uB85C\uADF8\uC778
6
7message.user.login.language.en=\uC601\uC5B4
8message.user.login.language.ko=\uD55C\uAE00
9
10# getBoardList.jsp
11message.board.list.mainTitle=\uAC8C\uC2DC\uAE00 \uBAA9\uB85D
12message.board.list.welcomeMsg=\uB2D8! \uAC8C\uC2DC\uD310\uC5D0 \uC624\uC2E0\uAC78 \uD658\uC601\uD569\uB2C8\uB2E4.
13message.board.list.search.condition.title=\uC81C\uBAA9
14message.board.list.search.condition.content=\uB0B4\uC6A9
15message.board.list.search.condition.btn=\uAC80\uC0C9
16message.board.list.table.head.seq=\uBC88\uD638
17message.board.list.table.head.title=\uC81C\uBAA9
18message.board.list.table.head.writer=\uC791\uC131\uC790
19message.board.list.table.head.regDate=\uB4F1\uB85D\uC77C
20message.board.list.table.head.cnt=\uC870\uD68C\uC218
21message.board.list.link.insertBoard=\uC0C8\uAE00 \uB4F1\uB85D
```

```

index.jsp  messageSource_ko.properties  messageSource_en.properties  x  servlet-conte
1# login.jsp
2message.user.login.title=LOGIN
3message.user.login.id=ID
4message.user.login.password=PASSWORD
5message.user.login.loginBtn=LOG-IN
6
7message.user.login.language.en=English
8message.user.login.language.ko=Korean
9
10# getBoardList.jsp
11message.board.list.mainTitle=BOARD LIST
12message.board.list.welcomeMsg=! Welcome to my BOARD
13message.board.list.search.condition.title=TITLE
14message.board.list.search.condition.content=CONTENT
15message.board.list.search.condition.btn=Search
16message.board.list.table.head.seq=SEQ
17message.board.list.table.head.title=TITLE
18message.board.list.table.head.writer=WRITER
19message.board.list.table.head.regDate=REGDATE
20message.board.list.table.head.cnt=CNT
21message.board.list.link.insertBoard=Insert Board

```

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/8a7bd97f-57f0-418f-bcbc-1dfe460dc8aa/messageSource\\_ko.txt](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/8a7bd97f-57f0-418f-bcbc-1dfe460dc8aa/messageSource_ko.txt)

[https://s3-us-west-2.amazonaws.com/secure.notion-static.com/65806162-964e-423a-b571-2574648c457b/messageSource\\_en.properties](https://s3-us-west-2.amazonaws.com/secure.notion-static.com/65806162-964e-423a-b571-2574648c457b/messageSource_en.properties)

<!-- 다국어 설정, id는 고정된 id를 작성해야 한다. ResourceBundleMessageSource를 바라보게 하려면 id는 반드시 messageSource -->
 <beans:bean id="messageSource"
 class="org.springframework.context.support.ResourceBundleMessageSource">
 </beans:bean>

## login.jsp

```

<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ taglib prefix="spring" uri="http://www.springframework.org/tags" %>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title><spring:message code="message.user.login.title"/></title>
</head>
<body>
<!-- CSS only -->
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-iYQcZEFbkJA/

```

```
<center>  
    <h1 align="center"><spring:message code="message.user.login.loginBtn"/></h1>  
    <a href="/login.me?lang=en" class="btn btn-outline-info btn-lg px-4 me-sm-3 fw-bold">  
        <spring:message code="message.user.login.language.en"/>  
    </a>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&~  
    <a href="/login.me?lang=ko" class="btn btn-outline-info btn-lg px-4 me-sm-3 fw-bold">  
        <spring:message code="message.user.login.language.ko"/>  
    </a>  
</center>  
<br>  
<form action="/login.me" method="post"><!-- ./는 현재 내 위치 ../ 내 위치에서 한단계 위 위치 -->  
    <table border="1" cellpadding="0" cellspacing="0" align="center">  
        <tr>  
            <td><spring:message code="message.user.login.id"/></td>  
            <td><input type="text" name="id" value="{usr.id}"/></td>  
        </tr>  
        <tr>  
            <td><spring:message code="message.user.login.password"/></td>  
            <td><input type="password" name="password" value="{usr.password }"/></td>  
        </tr>  
        <tr>  
            <td colspan="2" align="center"><input type="submit" value="{spring:message code='message.user.login.loginBtn'}/>" /></td>  
        </tr>  
    </table>  
</form>  
  
<!-- JavaScript Bundle with Popper -->  
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.1/dist/js/bootstrap.bundle.min.js" integrity="sha384-u10kvxwVY5kfMNBILK2hRnQ" crossorigin="anonymous"></script>  
</body>  
</html>
```

**servlet-context.xml**

```
<!-- 다국어 설정, id는 고정된 id를 작성해야 한다. ResourceBundleMessageSource를 바라보게 하려면 id는 반드시 messageSource -->
<beans:bean id="messageSource" class="org.springframework.context.support.ResourceBundleMessageSource">
  <beans:property name="basenames">
    <beans:list>
      <beans:value>message.messageSource</beans:value> <!-- 스케일 정보와 확장자는 빼줌. 브라우저에 셋팅이 돼있기 때문에 -->
    </beans:list>
  </beans:property>
</beans:bean>

<!-- LocaleResolver 등록 -->
<beans:bean id="localeResolver" class="org.springframework.web.servlet.i18n.SessionLocaleResolver"></beans:bean>

<!-- LocaleChangeInterceptor 등록 -->
<interceptors>
  <beans:bean class="org.springframework.web.servlet.i18n.LocaleChangeInterceptor">
    <beans:property name="paramName" value="lang" />
  </beans:bean>
</interceptors>
```

면접관이 관심있는 건 제출한 포트폴리오를 가지고 질문을 던진다.

계속 질문을 하게끔 만들어야 한다.

그렇지 않으면 부수적인 질문만 던진다.

기본, 중간, 심화단계 각각 개발을 한다.

1, 3, 5, 10년차에 맡을 일들이 나누어져있다.

각 단계에서 일이 다르기에 심화 계발하는 사람이 시간을 벌 수 있다.

10년차가 기본단계만 할 수는 없다.

그런식으로 전체가 업그레이드가 되면 팀을 2개로 나눌 수 있다.

60~70%의 코드를 만드는 사람은 3~5년차.

경력이 많은 사람이 담당해야 할 일 자체가 무에서 유를 창조해야 하고, 그 코드를 만들기 위해 고민이 필요하다.

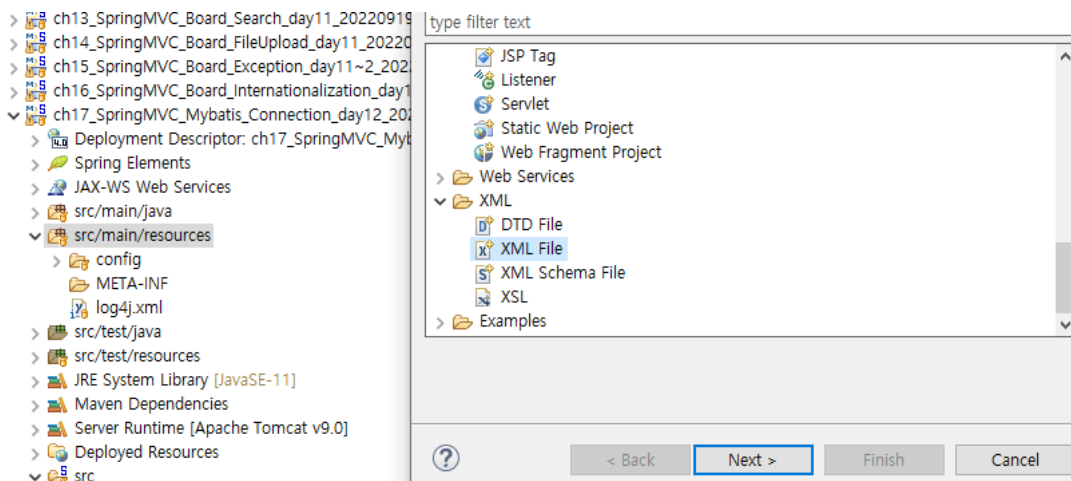
## 마이바티스 추가해주기

### pom.xml

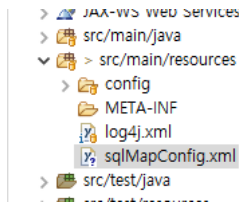
```
<!-- ch17_SpringMVC_Mybatis_Connection에서 추가, 마이바티스-->
<!-- https://mvnrepository.com/artifact/org.mybatis/mybatis -->
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis</artifactId>
  <version>3.5.6</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.mybatis/mybatis-spring, 마이바티스 연결해주는 것 -->
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis-spring</artifactId>
  <version>2.0.7</version>
</dependency>
```

### resources/



### sqlMapConfig.xml



## MyBatis - 마이바티스 3 | 시작하기

모든 마이바티스 애플리케이션은 SqlSessionFactory 인스턴스를 사용한다. SqlSessionFactory 인스턴스는 SqlSessionFactoryBuilder를 사용하여 만들 수 있다. SqlSessionFactoryBuilder는 XML 설정 파일에서 SqlSessionFactory 인스턴스를 빌드할 수 있다. XML 파일에서 SqlSessionFactory 인스턴스를 빌드하는 것은 매우 <https://mybatis.org/mybatis-3/ko/getting-started.html>



← → ↺ mybatis.org/mybatis-3/ko/getting-started.html



모든 마이바티스 애플리케이션은 SqlSessionFactory 인스턴스를 사용한다. S

다.  
XML 파일에서 SqlSessionFactory 인스턴스를 빌드하는 것은 매우 간단하다. 소

```
String resource = "org/mybatis/example/mybatis-config.xml";
InputStream inputStream = Resources.getResourceAsStream(resource);
SqlSessionFactory sqlSessionFactory = new SqlSessionFactoryBuilder
```

XML 설정 파일에서 지정하는 마이바티스의 핵심이 되는 설정은 트랜잭션을 제

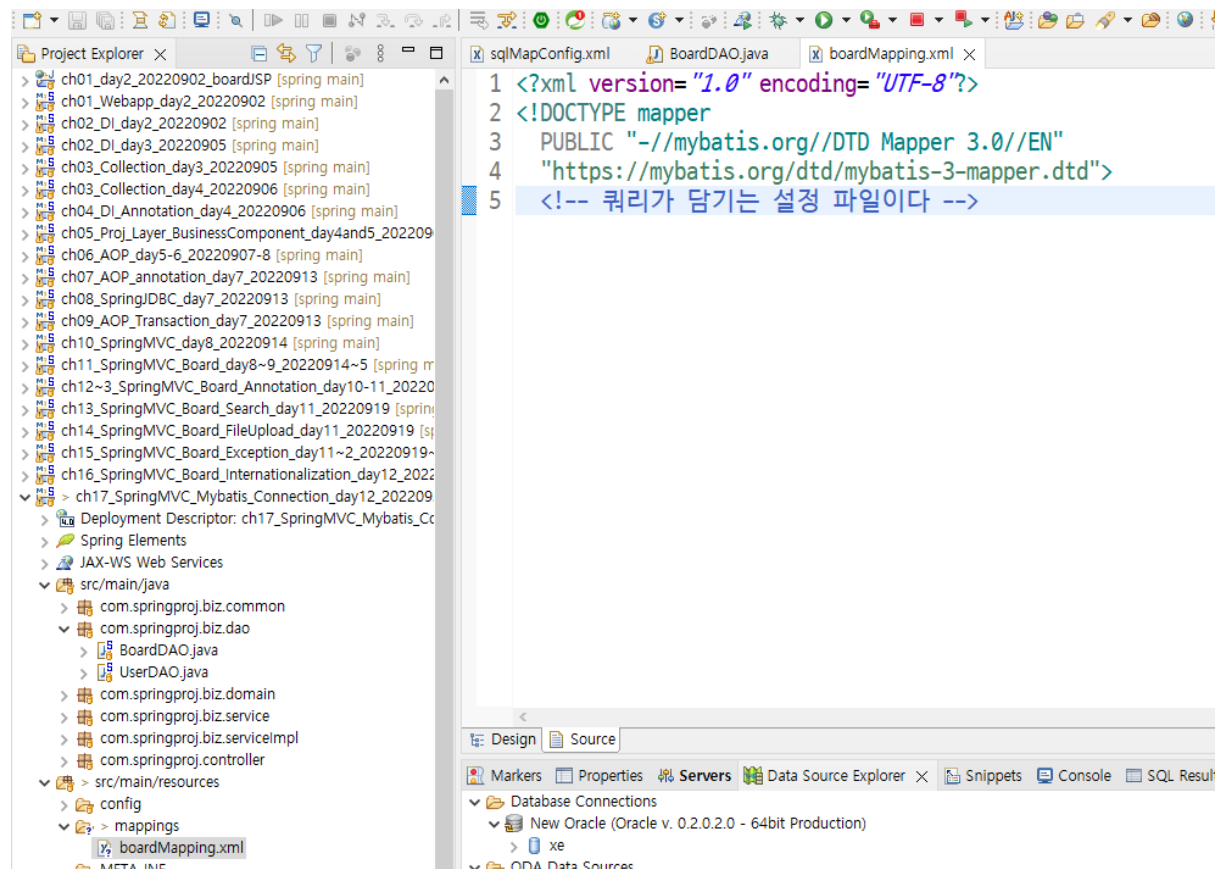
```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE configuration
  PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
  "https://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
  <environments default="development">
    <environment id="development">
      <transactionManager type="JDBC"/>
      <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="org/mybatis/example/BlogMapper.xml"/>
  </mappers>
```

```
sqlMapConfig.xml ×
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE configuration
3   PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
4   "https://mybatis.org/dtd/mybatis-3-config.dtd">
5
```

```
<!DOCTYPE configuration
  PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
  "https://mybatis.org/dtd/mybatis-3-config.dtd">
```

- SQL 명령어를 자바 코드에서 분리하여 XML 파일에 따로 관리.

## boardMapping.xml



## 매핑된 SQL 구문 살펴보기

이 시점에 SqlSession이나 Mapper클래스가 정확히 어떻게 실행되는지 해서는 충분히 이해하게 될 것이다.

위 예제처럼 구문은 XML이나 애노테이션을 사용해서 정의할 수 있다. : 보다 많은 기능이 추가되었다. SqlSession을 호출하는 XML 기반의 매퍼

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper
PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
"https://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="org.mybatis.example.BlogMapper">
```

## BoardDAO.java

```
}
import org.mybatis.spring.support.SqlSessionDaoSupport;
import org.springframework.beans.factory.annotation.Autowired
```

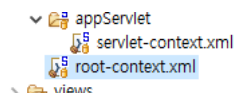


```
@Repository("boardDAO")
public class BoardDAO extends SqlSessionDaoSupport{
```

```
6 @Repository("boardDAO")
7 public class BoardDAO extends SqlSessionDaoSupport{ // DAO(Data Access Object)
8
9     public void setSqlSessionFactory(SqlSessionFactory sqlSessionFactory) {
0
```

```
5
7 import org.apache.ibatis.session.SqlSessionFactory;
3 import org.mybatis.spring.support.SqlSessionDaoSupport
```

root-context.xml는 business logic 기준으로 설정해주는 것. 평소에도 살려놓고 있어야 함. 제일 먼저 구동하고 있어야 함.



servlet-context.xml는 controller 기준으로 설정해주는 것.

root-context.xml

```
<!-- Spring과 Mybatis 연동 설정 -->
<bean id="sqlSession" class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="dataSource" ref="dataSource"></property> <!-- name="dataSource" Set호출하면서 ref="dataSource"자료를 전달한다 -->
    <property name="configLocation" value="classpath:sqlMapConfig.xml" />
    <!-- Mybatis설정에 대한 위치정보를 value="classpath:sqlMapConfig.xml"라고 알려주는 것 -->
</bean>
```

boardMapping.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">
<!-- 쿼리가 담기는 설정 파일이다 -->

<mapper namespace="Board">
    <insert id="insertBoard">
        <![CDATA[
            insert into board(seq, title, writer, content)
            values ((select nvl(max(seq),0)+1 from board), #{title}, #{writer}, #{content})
        ]]>
    </insert>
    <!-- '<', '>'는 비교 연산자로 인식하지 않고 여기에서는 꺾쇠로 본다. 설정을 <![CDATA[]]>로 --> <!-- // SQL문 - nvl: Null값이면 0으로 실행해라 -->
    <!-- ${}는 변수를 바로 가져온다. #{}은 실행되어지는 시점에 변수를 가져온다. 둘이 같다고 생각해도 상관없다. -->

    <select id="getBoardList_T">
        <![CDATA[
            select * from BOARD where title like '%'||#{searchKeyword}||'%' order by seq desc
        ]]>
    </select>
```

```

</select>

<select id="getBoardList_C">
<![CDATA[
    select * from BOARD where content like '%'||#{content}||'%' order by seq desc
]]>
</select>

<!-- List 보여주기 -->
<select id="getBoard">
<![CDATA[
    select * from BOARD where seq = #{seq}
]]>
</select>

<update id="updateBoard">
<![CDATA[
    update board set title = #{title}, content = #{content} where seq = #{seq}
]]>
</update>

<delete id="deleteBoard">
<![CDATA[
    delete from board where seq = #{seq}
]]>
</delete>

</mapper>

```

