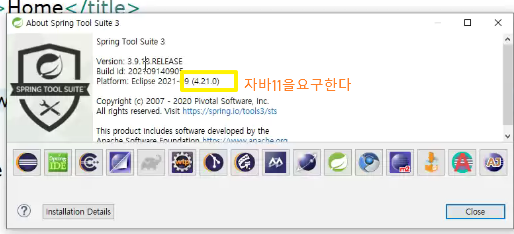
2024-02-23

* Bbs 게시판만들기 (~p36 까지함)

Spring은 자바가 없으면 의미가 없다

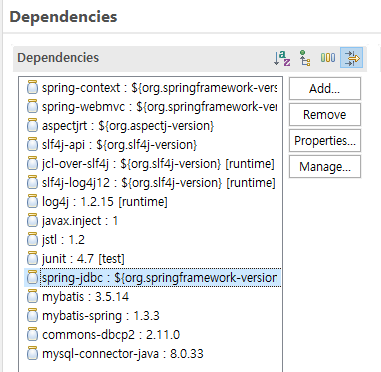
<myBatis> MyBatis3.X   
MyBatis는 JDBC를 기반으로 하는 SQL Mapping 프레임워크이다.  
SQL 쿼리를 프로그램 소스에서 분리해 관리한다

처음깔 때 오류나는 것은 톰캣이 없어서 그럼

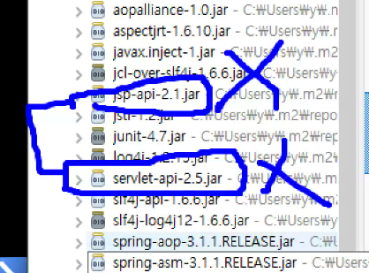


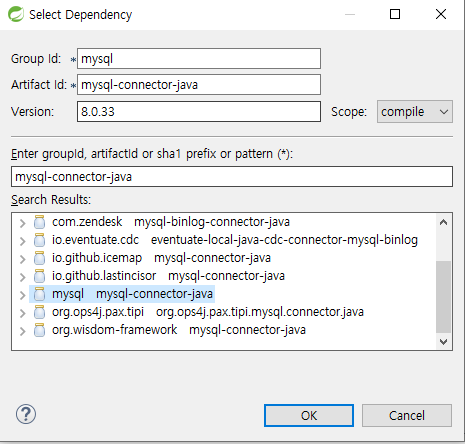
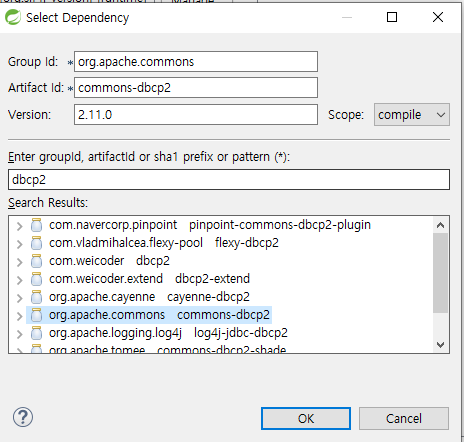
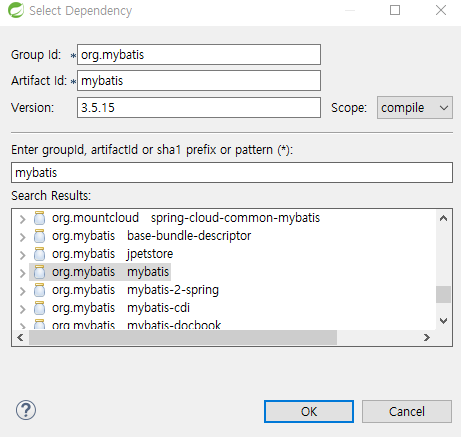
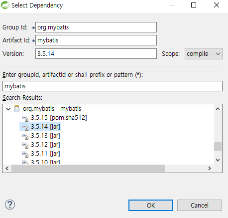
1. Build path 먼저 설정해준다(자바11, compiler 11확인, Project Facets 버전확인->runtime 톰캣 , Targeted Runtimes 확인, Web Project Settings 이름 프로젝트이름과 동일하게 줌)  
   -> 설정해줘야 이유 없는 오류들이 사라짐
2. Pom.xml : 의존라이브러리 설정  
   Overview, Dependencies 로 버전들 확인하고 추가할 것 추가하기  
   Artifact id와 project Name 확인  
   Framework 4.3.0. 이상으로 해야 postmappin , \_\_\_ 다 사용할 수 있다



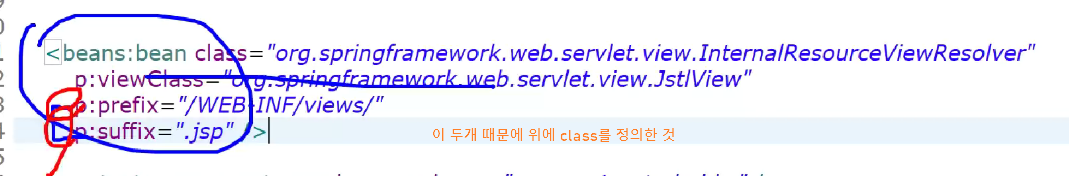


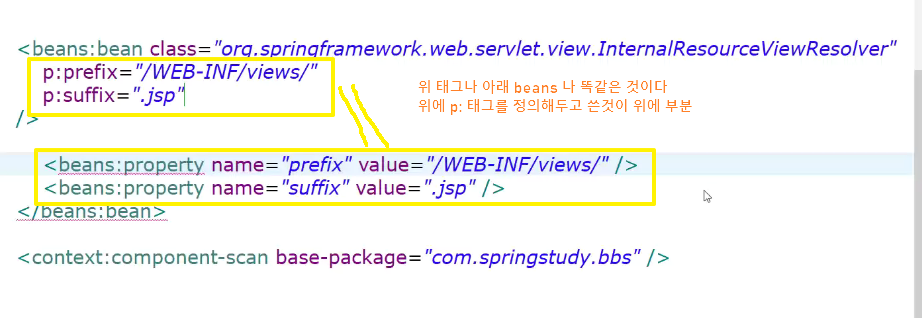
Dependencies

* Mybatis : 3.5.14
* Mybatis-spring : 1.3.3
* Spring-jdbc : springframework설정해준 버전(4.3.0이상)
* 삭제 api (충돌나기때문)

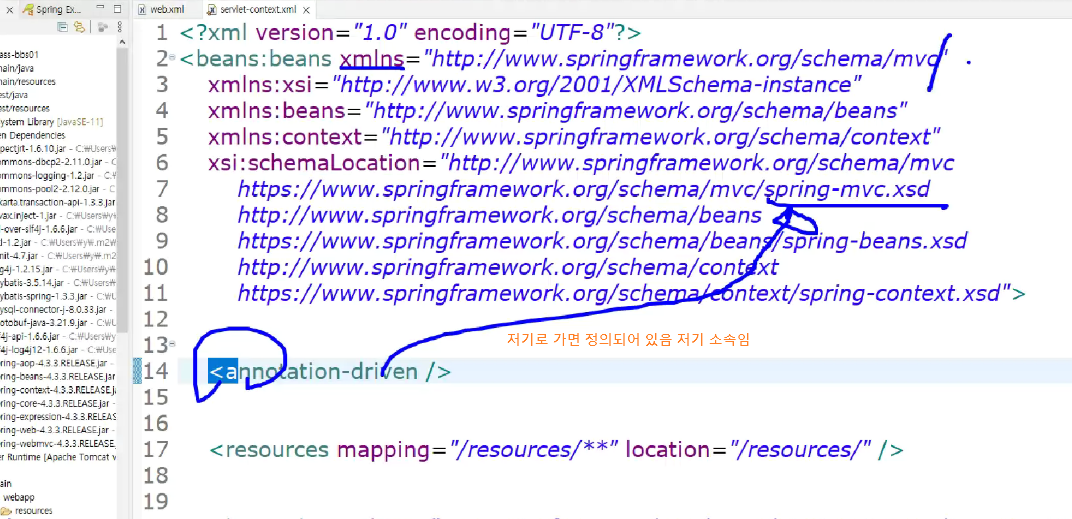
  
mybatis, 3점대 이상으로 설치

1. Web.xml  
   filter 부분 가지고 오기

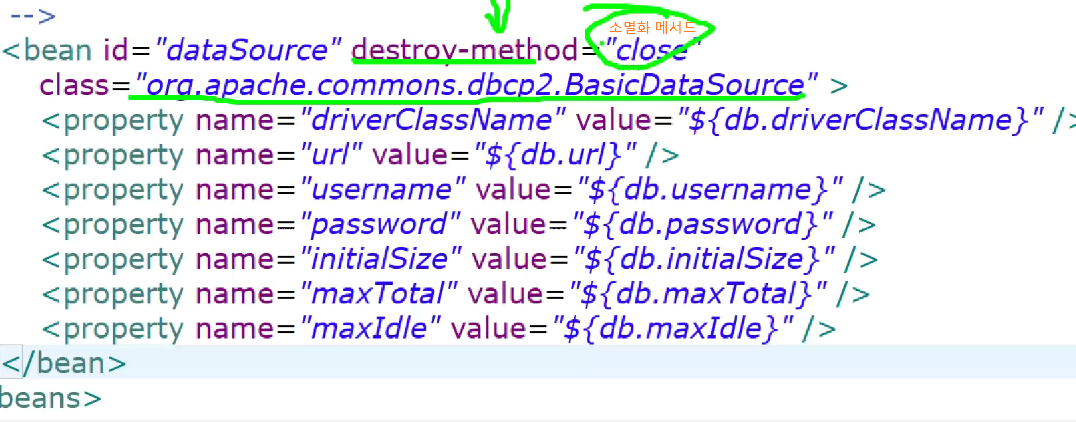


  
1)태그를 쓰는 방법  
2)속성을 쓰는 방법

1. Servlet-context.xml







---

**web.xml**->  
<param-value>  
/WEB-INF/spring/appServlet/servlet-context.xml  
톰캣실행될때 listener를 읽는다  
</param-value>

서블릿 클래스는 최초 요청이 들어올 때 톰캣 서버에 의해 서블릿 클래스가 로딩되고 인스턴스화 된다  
제일 먼저 servlet-context.xml이 실행!? -> bean 정의 파일

<load-on-startup>을 함으로 톰캣 실행되면서 초기화하도록 설정 -> 초기화 파라미터 읽는다 그러고 url 읽는다  
<url-pattern> url-pattern 지정 (/로?)

<filter> 인코딩용 필터 정의, 넘어오는 데이터를 문제 셋으로 처리된다 (utf-8) -> post방식의 요청은 문자셋처리 코드 작성하지 않아도됨  
-> 필터 : 서블릿 기술의 3대장이다 매핑에 의해서 실행 -> 필터 매핑 /\* 이렇게 한다 모든 요청이라는 의미

<resource-ref>

---  
MVC 설정  
**servlet-context.xml**

충돌막기 위해 xmlns:beans 사용  
xml 스키마 정의 파일 -> 스키마 위치를 줌 (https://www.springframework.org/schema/beans/ -> xml 정의한 것 등등)

<annotation-driven>  
MVC와 관련 있는 것들을 활성화 해줌

리소스로 매핑하겠다 리소스 밑에 찾아라 css , java , img는 다 resources 밑에 넣으면 됨

beans 태그로 bean을 준 것 (beans아래 bean이 있다는 것. Resolver

중복 없애기 위해서, 미리 정해주면 뒤에 이름만 반환하면 되니까 (이걸 써주지 않으면 이 거를 항상 써줘야 함)

<context:component-scan> package를 스캔한다는 것

board, index를 view-controller에서 지정해준 것을 읽는다 마지막에!?

---  
**view**에 jsp파일을 하나 만든다 -> boardList.jsp

controller를 만든다 -> 패키지 안에 클래스로 만든다 @controller , 열거형 (E) -> public String boardList를 줌 어노테이션으로 REquestMapping을 주는데 method는 get 방식으로 주고, 값은 여러개 줄거면 배열로 줌(value) -> return 할 때 String(문자열)으로 반환하면 뷰의 이름으로 간주한다 viewResolver 설정의 prefix와 suffix에 문자열로 반환, viewResolver에서 하나의 링크로 간주한다 /WEB-INF/views/ + boardList + .jsp 로 forward(포워딩)이 가능하다(그림으로 설명) ->

dao를 만든다

service를 만든다

domain을 만든다 -> 회원 게시판 정의하는

applicationContext -> DI 컨테이너 (web.xml 부분의 listener)

Dispacher 읽으면 servlet-context도 읽는다

---  
dataresource

**- src/main/resources/datasource/datasource.properties**maven에 sql있다 거기에 있는 정보다 => Driver  
##DBCP 사용할 때 정보

**- src/main/resources/datasource/datasource.xml**datasource -> 소멸화 메서드 close로 닫는다

---  
**- src/main/webapp/WEB-INF/spring/root-context.xml**

import하면 db정보를 끌어오는 것과 동일하다  
<import resource= "datasource/dbcpdatasource.xml /> 지정해줌

-1. 2.은 의존라이브러리 설정한것 말함

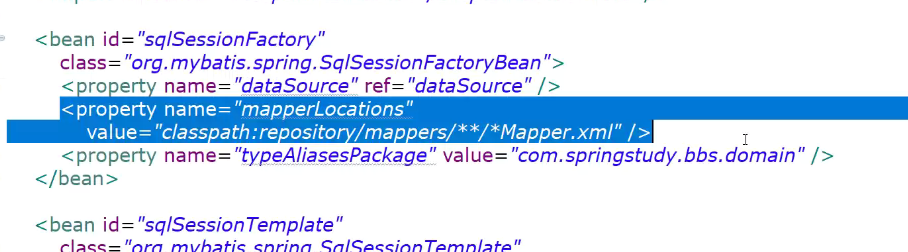
아래부분 namespace에 필요한 것을 설정할 수 있다(추가하고 제거가능) -> 여기에 없다는 것은 의존라이브러리가 없다는 것 => 없으면 pom.xml에 dependencies 에 추가하기 (spring-jdbc 추가함, jdbc가 tx에 의존하고 있어서 tx도 같이 받아진다)

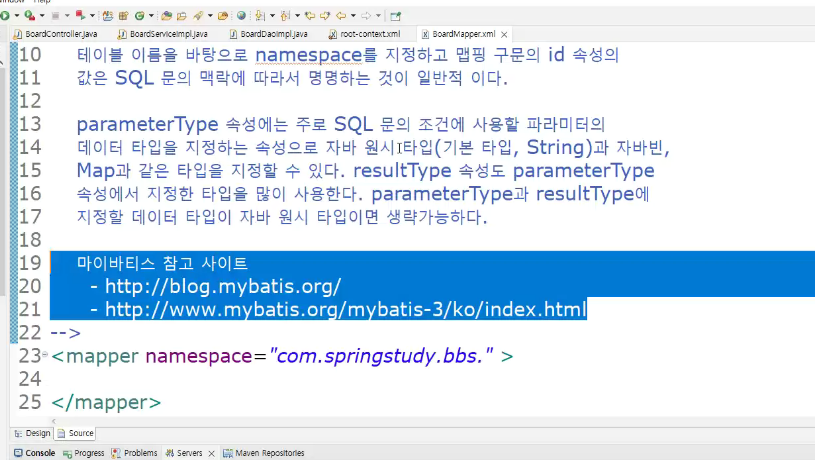
dataSource는 datasource> dbcpdatasource.xml 에 있는 것

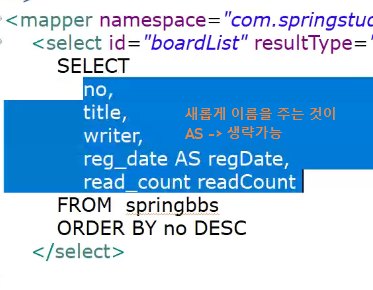
5. sqlSessionTemplate

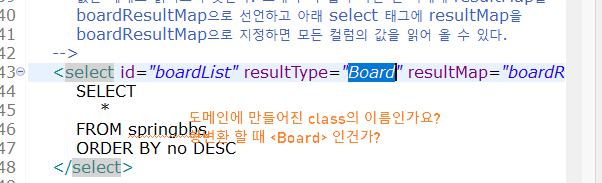
모든 정보 가지고 있는 것이 sqlSessionFactory이기 때문에  
c:sqlSessionFactory-ref="sqlSessionFactory" 이렇게 해줌   
-> 팩토리에 있는 매퍼 그런정보들을 가지고 있는 것 그것을 가지고 사용하면 됨

Maven dependencies 중에 mybaits 안에 있는 org.mybatis.spring.SqlSessionFactoryBean을 spring bean으로 정의해서 sqlsessionfactory를 생성한다.









type이 board인이유는 class이름을 그것으로 설정해줘서

그리고 board만 써줄 수 있는 이유는

- root-context에

<property name="typeAliasesPackage" value="com.springstudy.bbs.controller.domain"/>

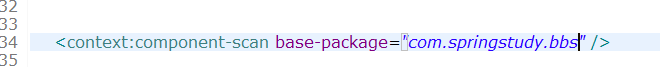
이 부분이 있어서 board만 써줘도 앞부분을 읽을 수 있는 것

@Autowired가 있으면 그것을 주입해 줄 것인데

인터페이스니까 그것이 포함되어 있는 class로 가줘야 함

@Service -> bean 정의한것

xml해석할 때 문제발생

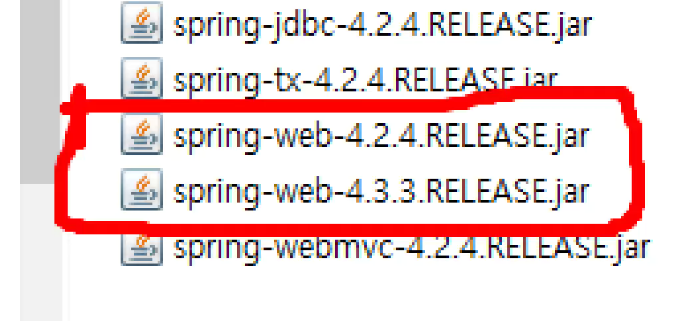
  
bbs 파일을 변경하면서 bbs로 설정되어 있는 것이 controller 로 변경된 것이다

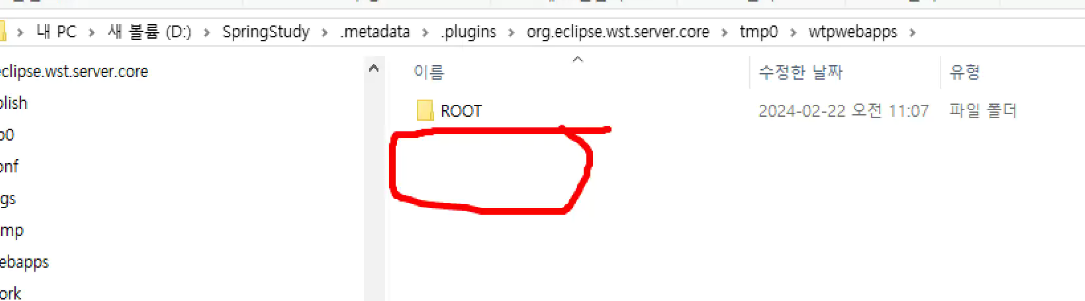
그러니까 controller만 인식하고 다른 bao, service 등이 읽어지지 않은 것

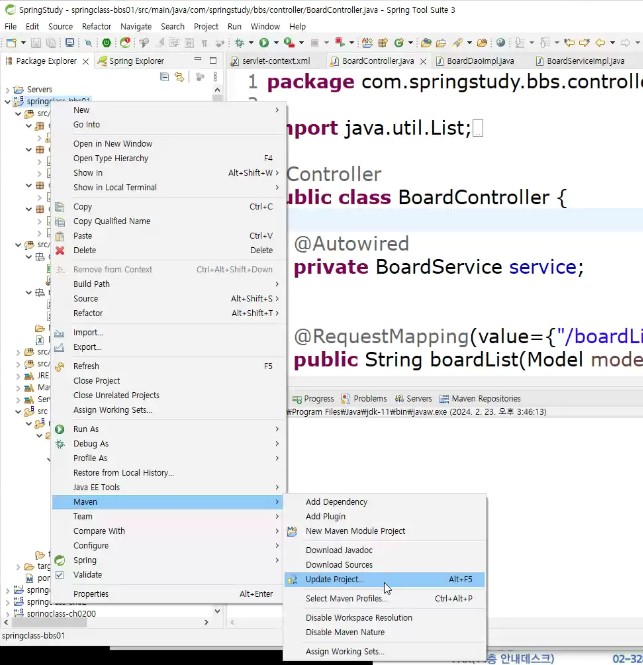
어노테이션 붙여도 , 의미가 없어진 것 , 스캔 받을 대상에서 벗어난 것

임시로 파일 업로드 한 상태에서 clean을 하면 그 저장되어 있는 것이 날라간다, clean 하기전에 파일들 수동으로 업로드 해야 됨

* Clean 가능한 상태인지 확인하고 clean하기

  
spring-web파일이 2개라서 발생한 문제다.

  
clean 하면 지워진다

  
정 안먹을 때 하는 건데 거의 건들지 않는 것이 좋음!

오후

try catch 만드는 것 sql로 간단하게 가지고 올 수 있음

---------

root-context.xml

실제 우리가 원하는 자바객체와 Mapper 맵핑을 시켜줄 수 있는 것을 typeAliasesPackage를 통해서 했다(property)

<--! 트랜잭션 처리 (Transaction) -->

DBCP를 사용하기 위한 datasource를 setter 주입 받도록 설정

class="org.springframework.jdbc.datasource.DataSourceTransactionManager" -> 트랜잭션 매니저

SQL부분 완성되어 있는 파일을 Mysql에 끌고와서 select까지 shift+ctrl+enter 해서 생성해주기

---

com.springstudy.bbs.domain > Board.java 클래스 파일 만들기 -> SQL에 생성한 부분을 참고해서 만들기 (varchar는 String으로, int 는 int로, TIMESTAMP는 Timestamp로 해주면 됨 ㅋㅋ -> 생성자 만들고 -> getter와 setter 만들기\_source를 이용하면 손쉽게 만들수 있음)

---

- com.springstudy.bbs.service.BoardService

service에는 interface로 BoardService 만들기 -> 반환타입 Board가 담겨있는 List로 만들었다 (public abstract 생략된것 interface니까ㅏ(?))

- com.springstudy.bbs.service.BoardServiceImpl

@Service

//비즈니스 로직을 처리하는 객체 - 필요한 데이터는 BoardDao를 이용해서 가져온다 -> 주입하는것

private BoardDao boardDao;

//페이징처리

---

- com.springstudy.bbs.dao.BoardDao

게시글 리스트와 상세정보 가지고 오는 method를 정의

getBoard-> domain에서 생성된 것

- com.springstudy.bbs.dao.BoardDaoImpl > BoardDao를 상속하여 만들기

//SqlSessionTemplate(이 아이를 의존하고 있다)를 사용해서 DB에 접근해서 데이터를 읽어오는 클래스

//저장소임 @Repository (사실 component로 해도 되지만 정확하게 어떤 역할을 하는지 알려주기 위해서 Repository로 주는 것이다)

private SqlSessionTemplate sqlSession; -> 여기에 주입한 것임! DB에 연결하기 위해 필요하기 때문에

sqlSessionTemplate는 root-context.xml에 있음

public List<Board> boardList() { return sqlSession.selectList("") } => 한개이면 select, 여러개면 selectList. -> 반환타입에 맞춰서 반환해줌

\* 이 예제는 MyBatis가 제공하는 SqlSessionTemplate 객체를 사용하기

\* 때문에 스프링으로부터 DI 받을 수 있도록 생성자나 setter를 준비해야 한다.

---

- com.springstudy.bbs.controller.BoardController

위에서 controller만들었다

Service만든 것을 주입!

JVM에 사용하는 메모리영역에 데이터 저장하고 forward해서 가지고 꺼내서 쓰면됨 -> 내장객체인 request였음

-> 저장하는 방법 모델이라는 클래스를 제공함 (갔다가 꺼내 쓸 수 있게 함)

@RequestMapping(에 붙어있는 method에 Model model)을 알려주면 디스패쳐가 인식한다

1. 입력 처리

2. 요청에 대한 처리를 하고 결과 데이터 - 게시 글 리스트 - 모델(model)

List<Board> bList = service.boardList();

2.5. 결과 데이터 만드는 것

3. 뷰에 결과를 출력

return "boardList";

---

- src/main/resources/repository/mappers/BoardMapper.xml

⭐namespace 가 중요하다, 충돌발생을 줄임

게시글 리스트를 가져오는 맵핑 구문

<resultMap> -> 결과 맵이니까 DB와 연결 , 여러개 만들 수 있음 (테이블의 컬럼과 도메인 객체의 프로퍼티를 설정하는 것)

dao로 가지고 가서 상수로 만든다 =>com.springstudy.bbs.mapper.BoardMapper 이부분

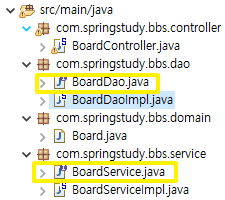
dao에 sqlSession에 추가해주는 부분을 추가해준다 (그 외에는 mybatis가 다 해준다)

DB에 이름이 reg\_date이기 때문에 그것을 객체로 가지고 올 때는 그 이름이 여기 에 없다 그래서

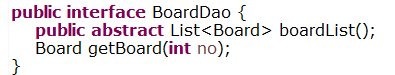
<result property="regDate" column="reg\_date" />

<result property="readCount" column="read\_count" />

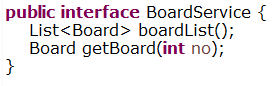
이렇게 해줌으로 spring에서는 regDate로 사용하겠다고 알려주는 것이 resultMap이다.



* BoardDao.java



* BoardService.java



⭐BoardDao와 BoardService부분의 boardList()는 각자 다른 리스트이다.  
BoardDao는 쿼리문을 포함한 리스트이고, BoardService는 BoardServiceImpl에 넣어줄 비즈니스 로직 처리에 대한 것을 받은 리스트이다.

* BoardDaoImpl.java



* BoardMapper.xml의 부분인 SQL의 쿼리를 나타내는 id=”boardList가 위에 불러지는 것.

