[ 14 ] Mybatis - I

처음 JSP&Servlet을 이용할 때는 많은 코드양으로 DB에 접근하다가, Spring에서는 JDBC Template을 이용해서 코드양을 많이 줄일 수 있었다. Mybatis를 이용하면 코드를 더 줄일 수 있고, 자바 코드가 아닌 xml 파일로 DB에 엑세스한다. MyBatis는 DB접근을 자바가 아닌 xml로 대체하도록 하는 라이브러리이다. SQL 코드를 자바코드와 분리시키는데 목적이 있다.

진행순서

1. pom.xml:필요라이브러리 의존추가

2. web.xml :한글처리. \*.do

3. 실행 첫화면 : css 추가해서 테스트

4. db.properties : DB환경설정정보

5. root-context.xml에 dataSource 빈생성

6. DTO

7. xml 생성 :mapper(empList,deptList) + config

8. root-context.xml에 SqlSessionfactoryBean, sessionTemplate(SqlSessionTemplate) 빈생성

9. DAO

10. Service

11. Controller

12. emp.jsp

Dao –> Jdbc template

-> MyBatis (DAO / interface)

-> JPA

1. **pom.xml**

<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>5.2.8.RELEASE</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.mybatis/mybatis -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>3.5.5</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.mybatis/mybatis-spring -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis-spring</artifactId>

<version>2.0.5</version>

</dependency>

lombok

<!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->

<dependency>

<groupId>org.projectlombok</groupId>

<artifactId>lombok</artifactId>

<version>1.18.24</version>

<scope>provided</scope>

</dependency>

1. **web.xml**

<filter>

<filter-name>encodingFilter</filter-name>

<filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>

<init-param>

<param-name>encoding</param-name>

<param-value>UTF-8</param-value>

</init-param>

<init-param>

<param-name>forceEncoding</param-name>

<param-value>true</param-value>

</init-param>

</filter>

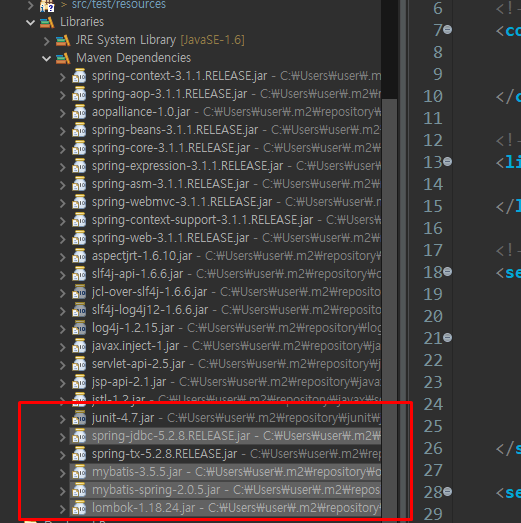
<filter-mapping>

<filter-name>encodingFilter</filter-name>

<url-pattern>/\*</url-pattern>

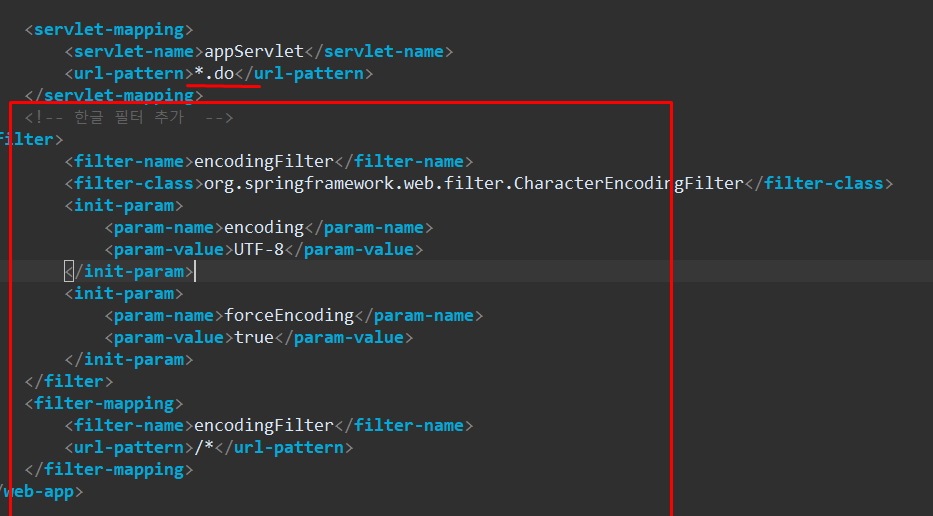
</filter-mapping>

* Maven 확인



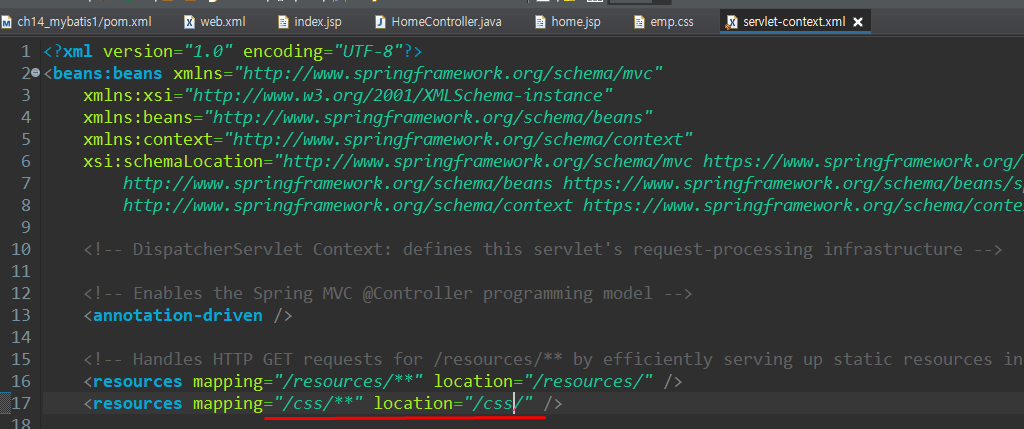
wep.xml

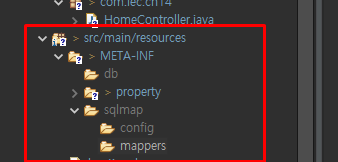
(한글처리) , \*.do



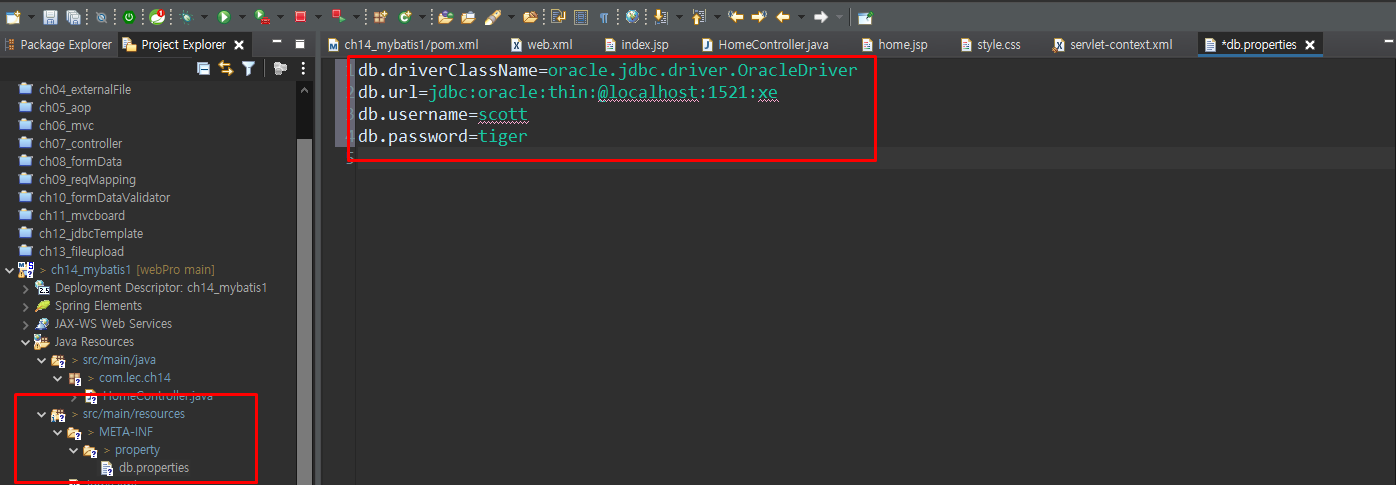
Index.jsp 생성

1. Css





1. db 환경설정정보



**src/main/resources/META-INF/property/db.properties**

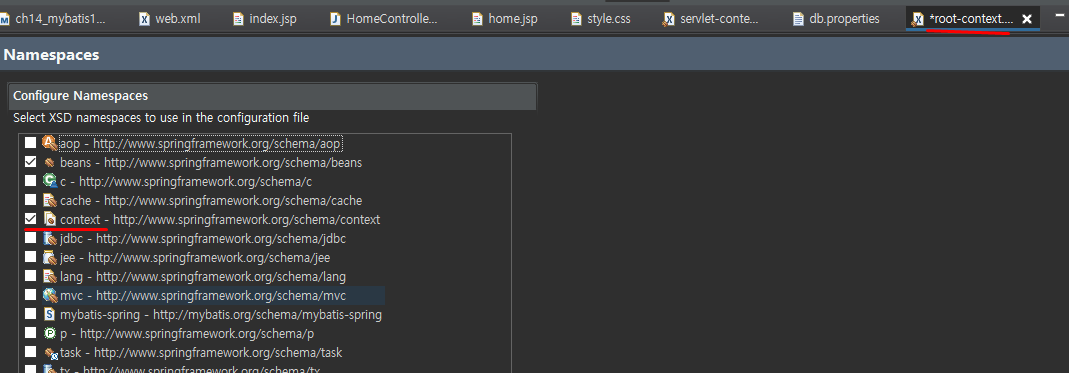
db.driverClassName=oracle.jdbc.driver.OracleDriver

db.url=jdbc:oracle:thin:@localhost:1521:xe

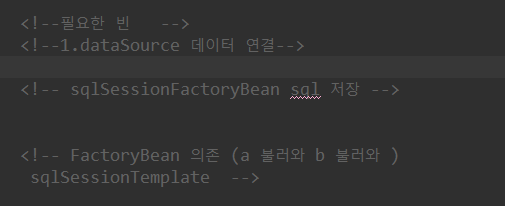
db.username=scott

db.password=tiger

1. root-context.xml에 dataSource 빈생성

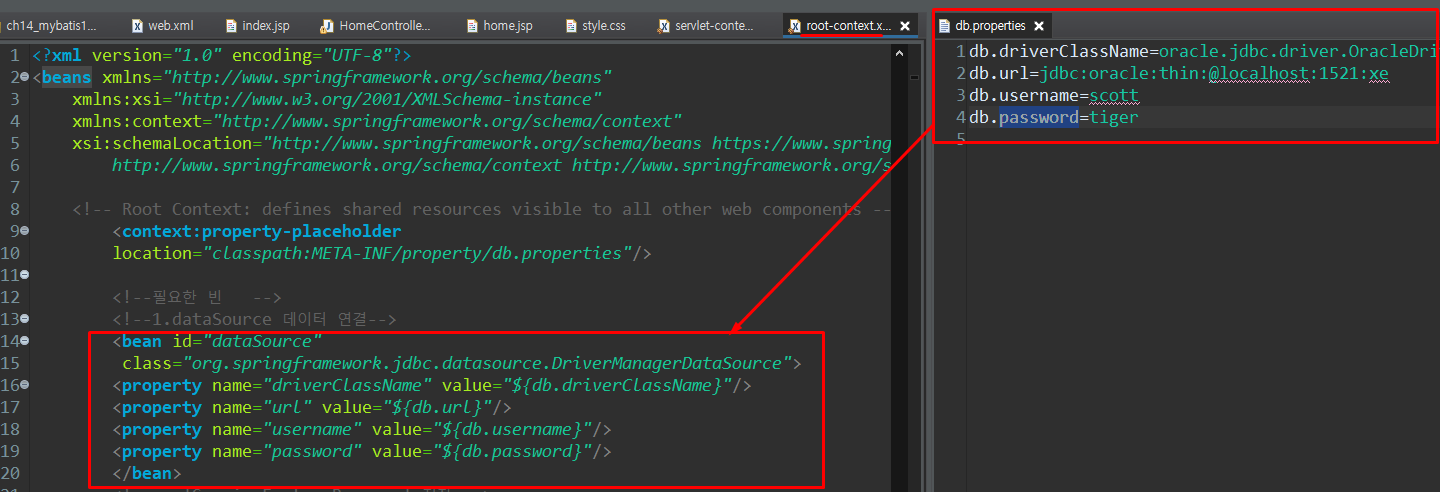


필요한 bean

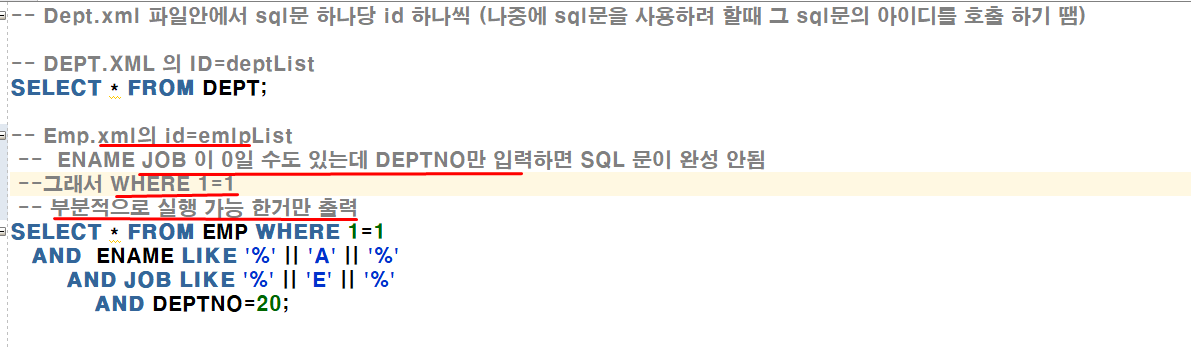


-DataSourc

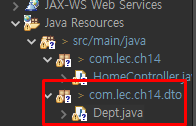
Class = DriverManagerDataSource 주소



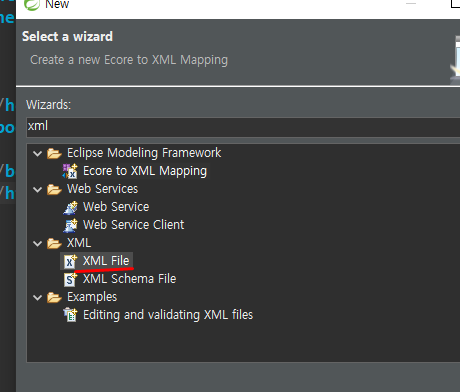
SQL 작성



1. DTO 생성(반드시 sql 과 변수명 같게 !)

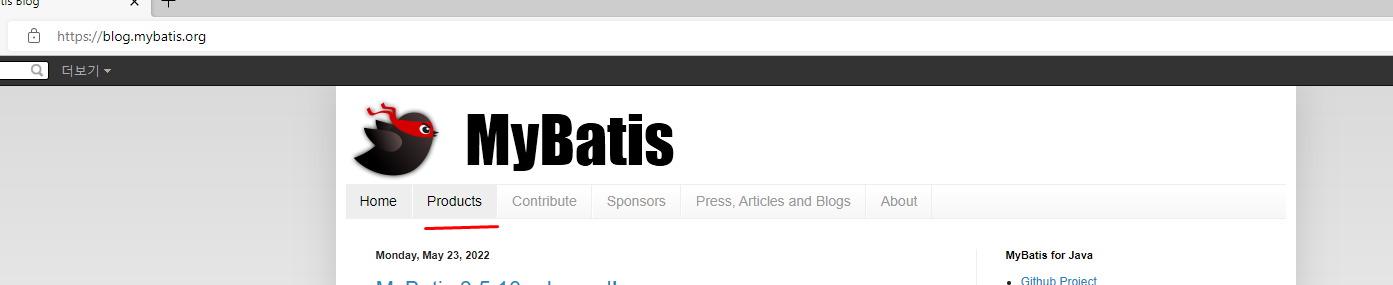


1. xml 생성 :mapper(empList,deptList) + config

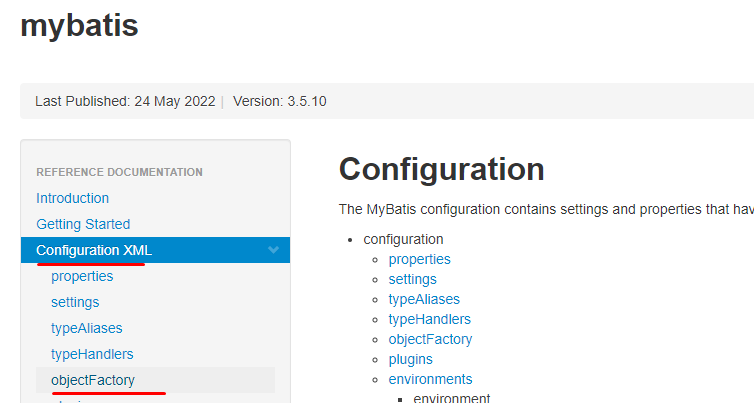




https://mybatis.org/





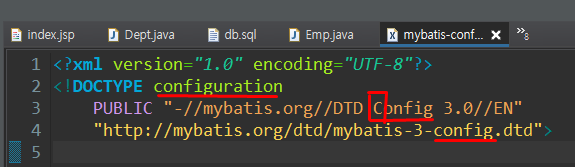


<!DOCTYPE mapper

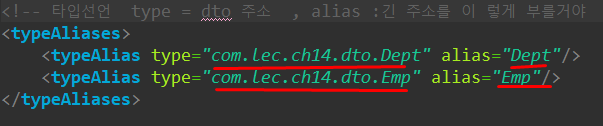
PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

Mybatios-config 와서 복붙 후 수정



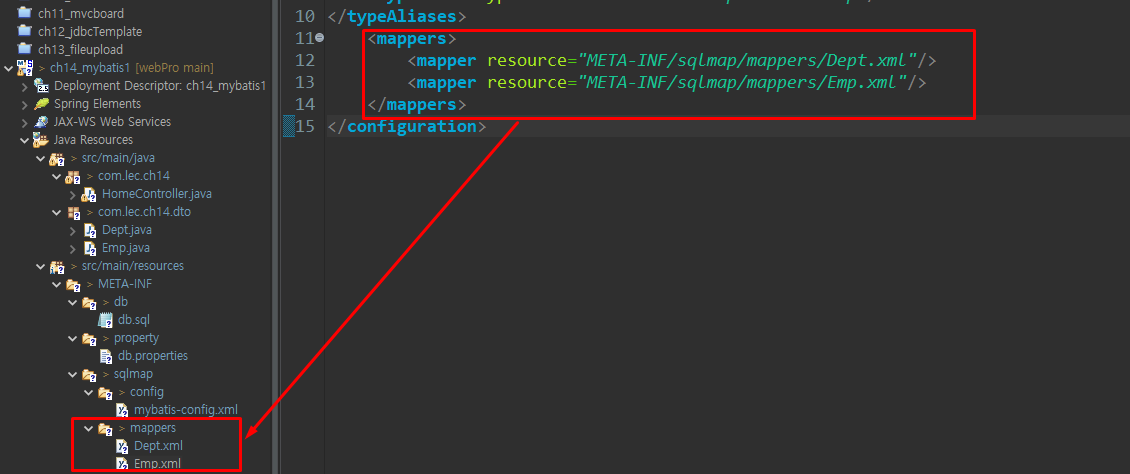
-타입 선언



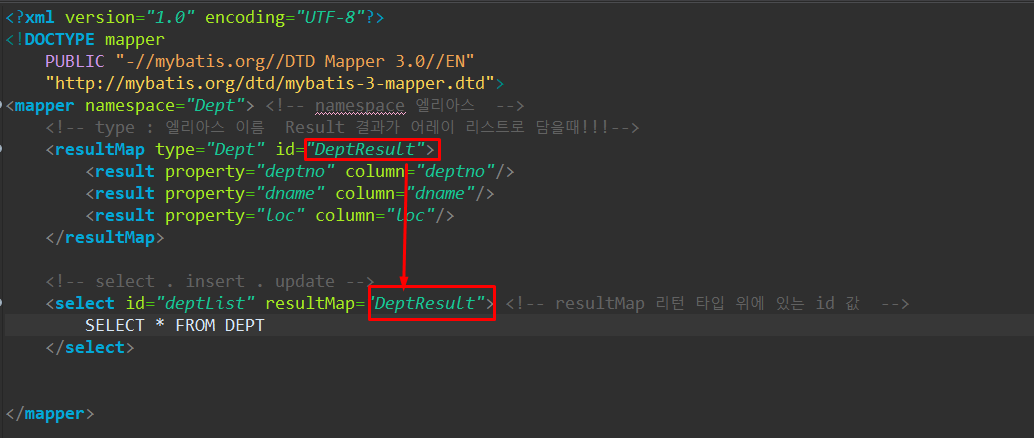
Type : dto 주소

Alias : 이 주소를 이렇게 간략하게 부를거야

Mappers 밑에 xml 파일 생성



Dept.xml

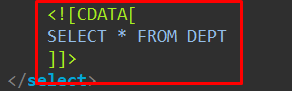


<resultMap type = 원래는 주소 다적는건데 엘리아스 사용하여 간단하게 입력>

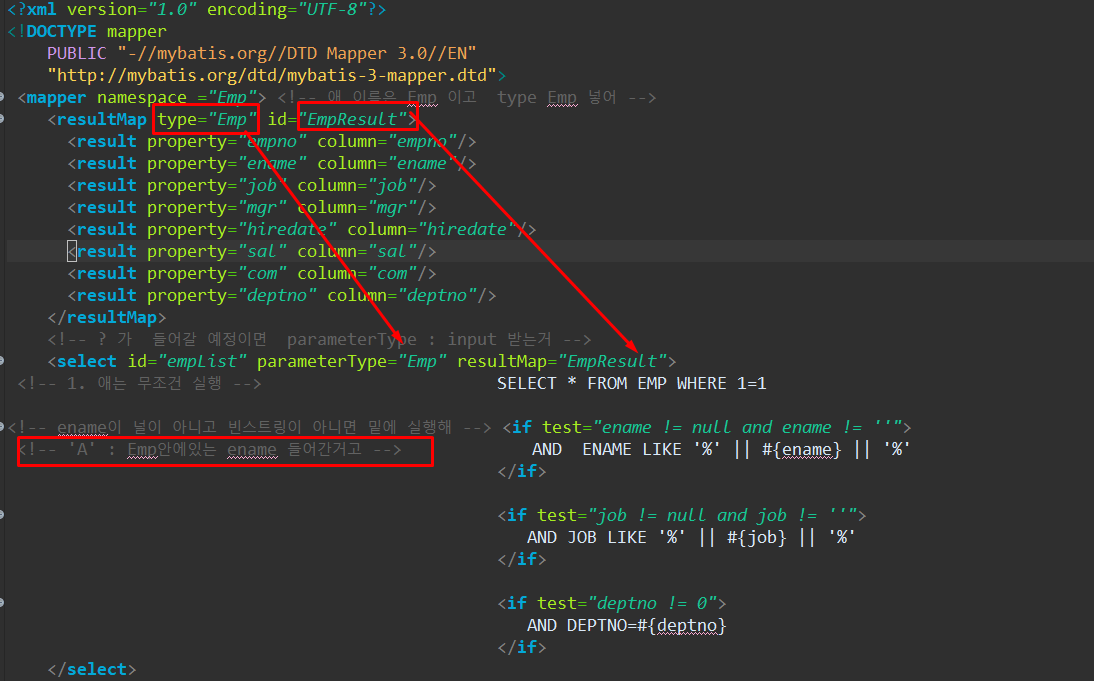
<select, insert , update>

Id 는 deptList 이고 리턴 타입은 위에서적은 id 값 리턴

-sql 실행 안될 때 ?



Emp.xml

<!**DOCTYPE** mapper

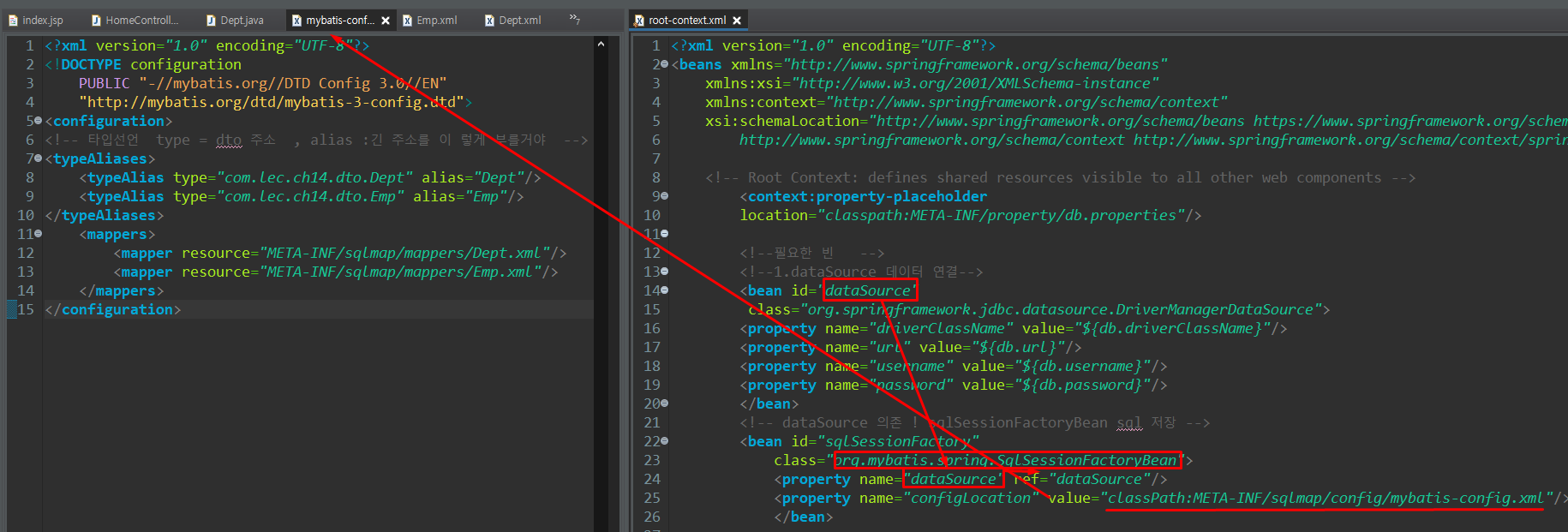
<!**DOCTYPE** mapper

PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"

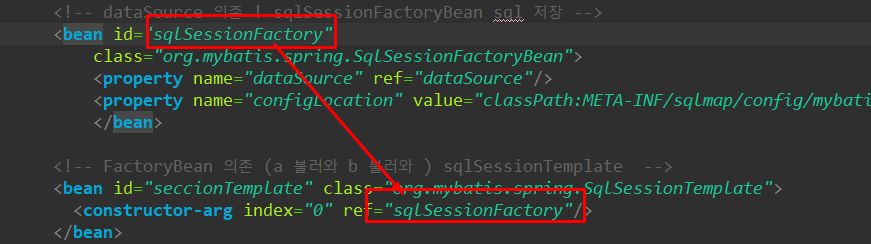
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

-Root-context.xml

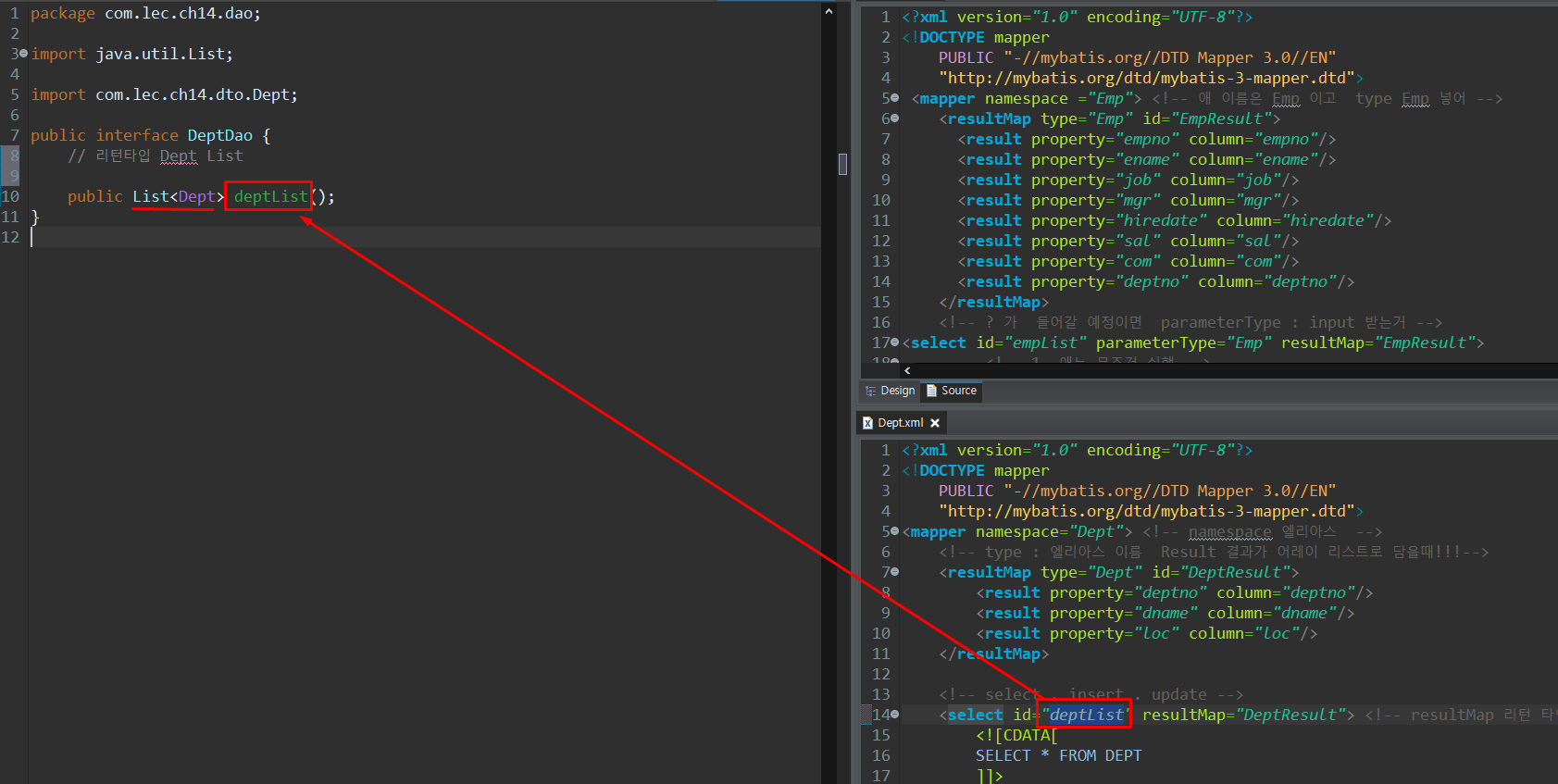
sqlSessionFactoryBean (sql저장 장소) dataSource 의존



-SeccionTemplate (sqlSessionFactory 의존)



-DeptDao 인터페이스 생성

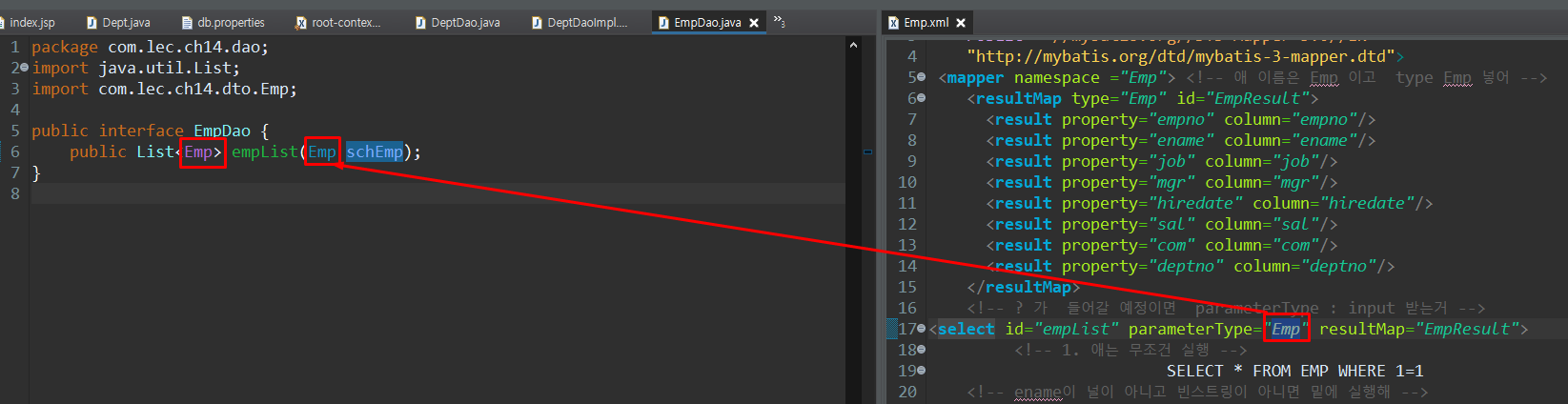


Public 리턴 타입은 Dept에List (java.util.List)

DeptList id 값과 같게

-implements DeptDao 받은 DeptDaoImpl 생성

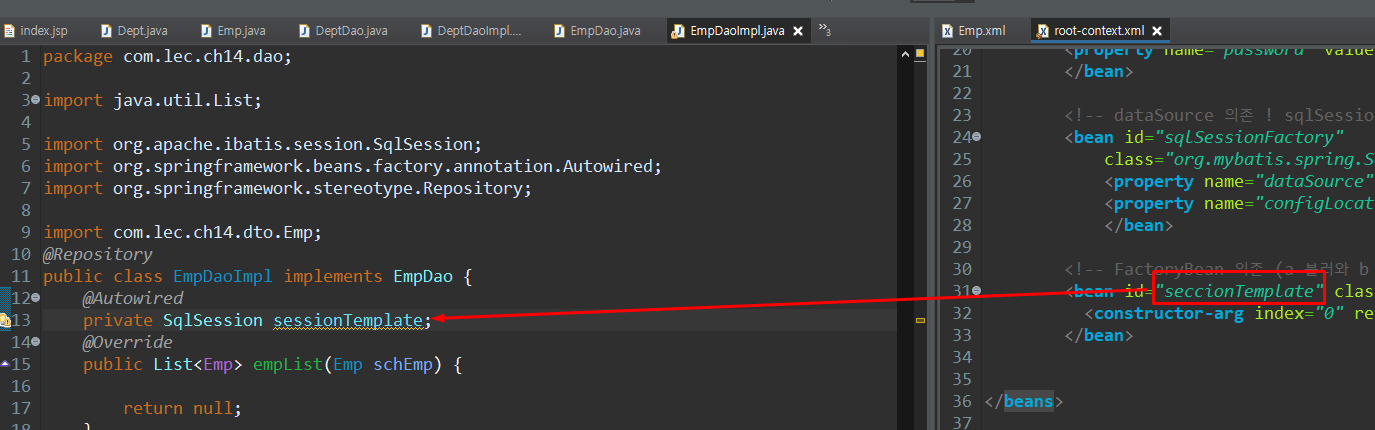
EmpDao

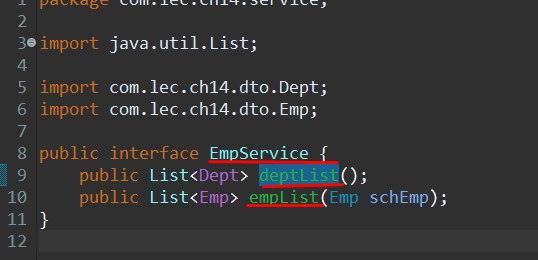


<Emp>dto

Emp 는 paremeterType 에서의 EMP

-EmpDaoImpl



-EmpService 인터페이스 생성 

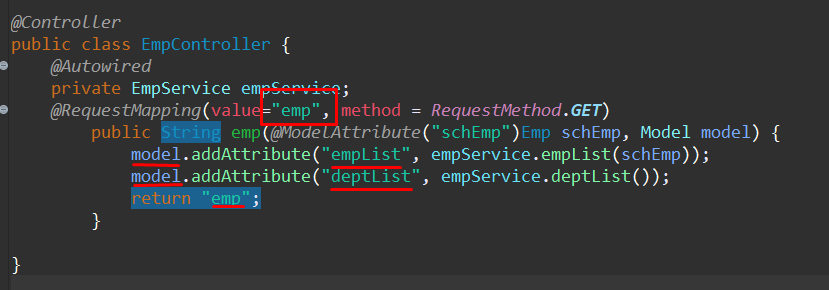
Dept 에 deptList

Emp 에 empList 함수 호출

- implements 받는 EmpServiceImpl 생성



-EmpController 생성



-Model 이용해 empList , deptList 연결

Return “emp.jsp” 로