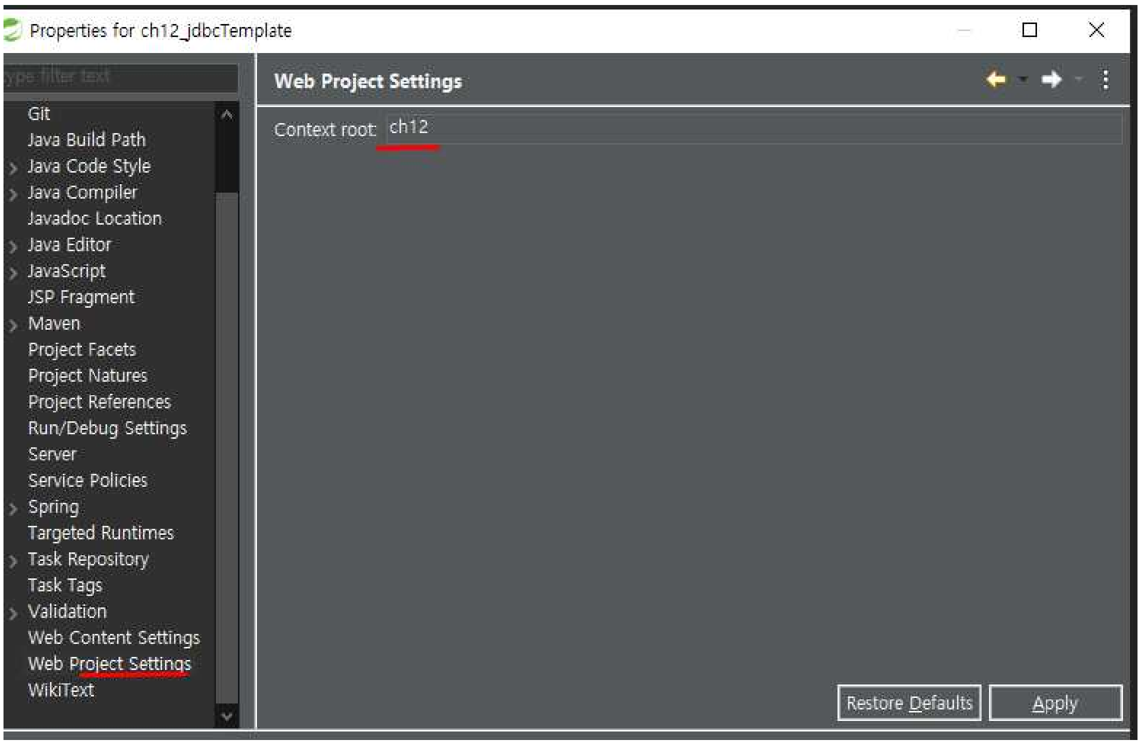
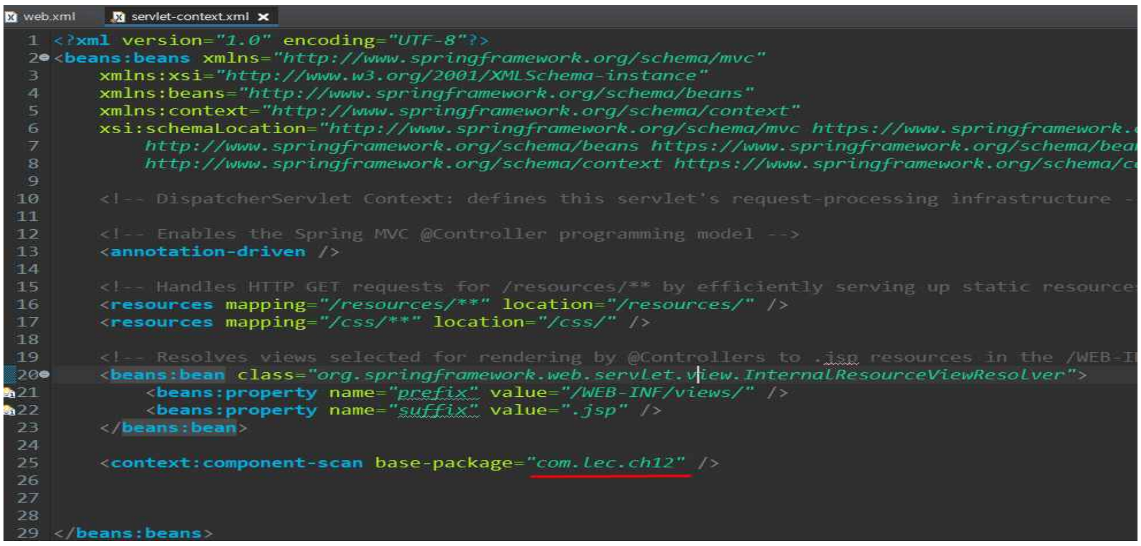
ch11 닫고 복붙 ch12 - properties -에서 변경



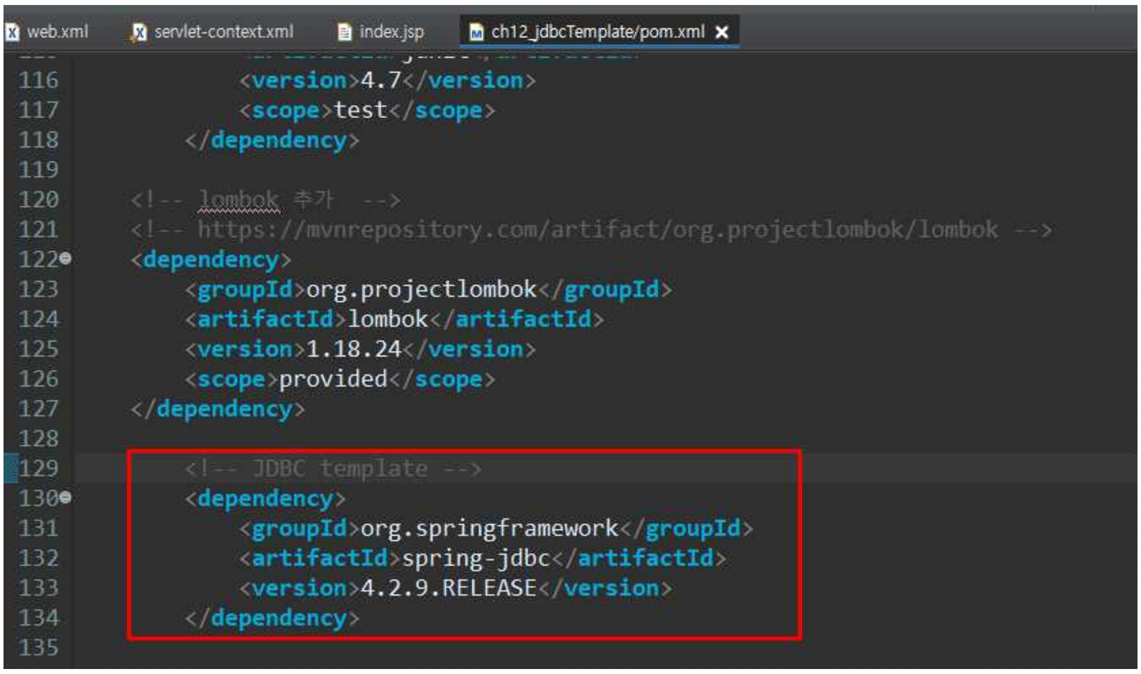
Context root: ch12

- ch12로 변경 했으니 servlet-context.xml에서 ch12 로 변경



```
<?xml version="1.0" encoding="UTF-8"?>
<beans:beans xmlns="http://www.springframework.org/schema/mvc"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:beans="http://www.springframework.org/schema/beans"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/mvc https://www.springframework.
        http://www.springframework.org/schema/beans https://www.springframework.org/schema/bea
        http://www.springframework.org/schema/context https://www.springframework.org/schema/c

    <!-- DispatcherServlet Context: defines this servlet's request-processing infrastructure -

    <!-- Enables the Spring MVC @Controller programming model -->
    <annotation-driven />

    <!-- Handles HTTP GET requests for /resources/** by efficiently serving up static resource
    <resources mapping="/resources/**" location="/resources/" />
    <resources mapping="/css/**" location="/css/" />

    <!-- Resolves views selected for rendering by @Controllers to .jsp resources in the /WEB-I
    <beans:bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">
        <beans:property name="prefix" value="/WEB-INF/views/" />
        <beans:property name="suffix" value=".jsp" />
    </beans:bean>

    <context:component-scan base-package="com.lec.ch12" />


</beans:beans>
```

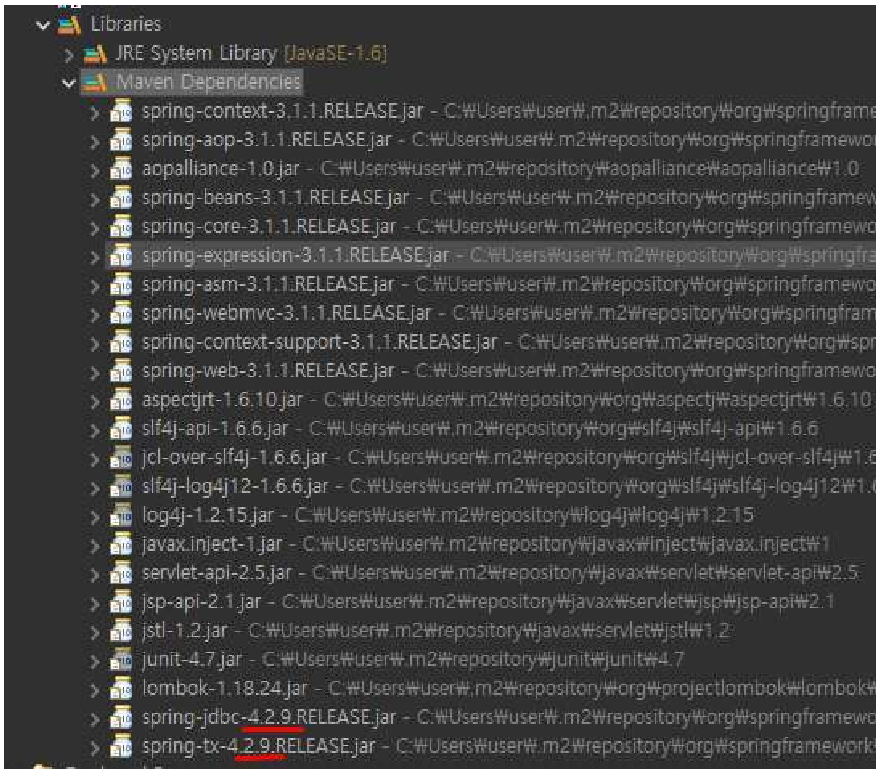
- porm.xml 에 추가



```
            <version>4.7</version>
            <scope>test</scope>
        </dependency>

    <!-- lombok 추가  -->
    <!-- https://mvnrepository.com/artifact/org.projectlombok/lombok -->
    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <version>1.18.24</version>
        <scope>provided</scope>
    </dependency>

        <!-- JDBC template -->
        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-jdbc</artifactId>
            <version>4.2.9.RELEASE</version>
        </dependency>

```

```
<!-- JDBC Template 둘 중에 하나 -->
<dependency>
                <groupId>org.springframework</groupId>
                <artifactId>spring-jdbc</artifactId>
                <version>4.1.4.RELEASE</version>
</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-jdbc -->
                <dependency>
                    <groupId>org.springframework</groupId>
                    <artifactId>spring-jdbc</artifactId>
                    <version>4.2.9.RELEASE</version>
                </dependency>
```

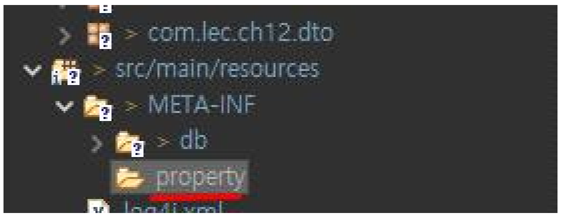
-Maven에서 확인

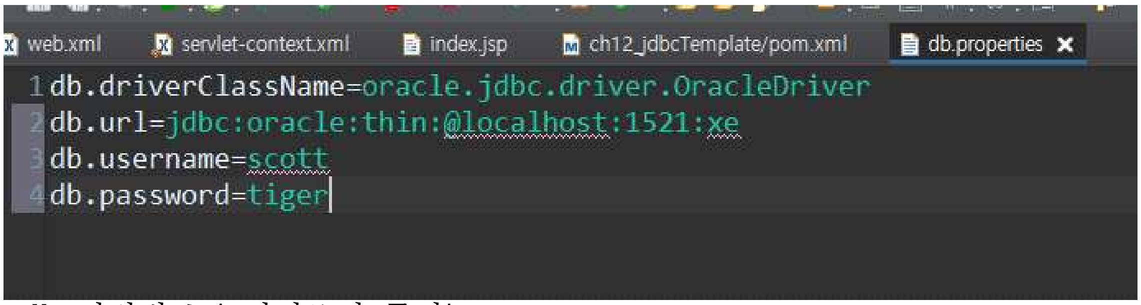


```
Libraries
  JRE System Library [JavaSE-1.6]
  Maven Dependencies
    spring-context-3.1.1.RELEASE.jar - C:₩Users₩user₩.m2₩repository₩org₩springframe
    spring-aop-3.1.1.RELEASE.jar - C:₩Users₩user₩.m2₩repository₩org₩springframewo
    aopalliance-1.0.jar - C:₩Users₩user₩.m2₩repository₩aopalliance₩aopalliance₩1.0
    spring-beans-3.1.1.RELEASE.jar - C:₩Users₩user₩.m2₩repository₩org₩springframew
    spring-core-3.1.1.RELEASE.jar - C:₩Users₩user₩.m2₩repository₩org₩springframewo
    spring-expression-3.1.1.RELEASE.jar - C:₩Users₩user₩.m2₩repository₩org₩springfra
    spring-asm-3.1.1.RELEASE.jar - C:₩Users₩user₩.m2₩repository₩org₩springframewo
    spring-webmvc-3.1.1.RELEASE.jar - C:₩Users₩user₩.m2₩repository₩org₩springfram
    spring-context-support-3.1.1.RELEASE.jar - C:₩Users₩user₩.m2₩repository₩org₩spr
    spring-web-3.1.1.RELEASE.jar - C:₩Users₩user₩.m2₩repository₩org₩springframewo
    aspectjrt-1.6.10.jar - C:₩Users₩user₩.m2₩repository₩org₩aspectj₩aspectjrt₩1.6.10
    slf4j-api-1.6.6.jar - C:₩Users₩user₩.m2₩repository₩org₩slf4j₩slf4j-api₩1.6.6
    jcl-over-slf4j-1.6.6.jar - C:₩Users₩user₩.m2₩repository₩org₩slf4j₩jcl-over-slf4j₩1.6
    slf4j-log4j12-1.6.6.jar - C:₩Users₩user₩.m2₩repository₩org₩slf4j₩slf4j-log4j12₩1.6
    log4j-1.2.15.jar - C:₩Users₩user₩.m2₩repository₩log4j₩log4j₩1.2.15
    javax.inject-1.jar - C:₩Users₩user₩.m2₩repository₩javax₩inject₩javax.inject₩1
    servlet-api-2.5.jar - C:₩Users₩user₩.m2₩repository₩javax₩servlet₩servlet-api₩2.5
    jsp-api-2.1.jar - C:₩Users₩user₩.m2₩repository₩javax₩servlet₩jsp₩jsp-api₩2.1
    jstl-1.2.jar - C:₩Users₩user₩.m2₩repository₩javax₩servlet₩jstl₩1.2
    junit-4.7.jar - C:₩Users₩user₩.m2₩repository₩junit₩junit₩4.7
    lombok-1.18.24.jar - C:₩Users₩user₩.m2₩repository₩org₩projectlombok₩lombok₩
    spring-jdbc-4.2.9.RELEASE.jar - C:₩Users₩user₩.m2₩repository₩org₩springframewo
    spring-tx-4.2.9.RELEASE.jar - C:₩Users₩user₩.m2₩repository₩org₩springframework
```

-src/main/resources 에 무조건 밑에 있어야함

-property 밑에 파일 db.properties 생성





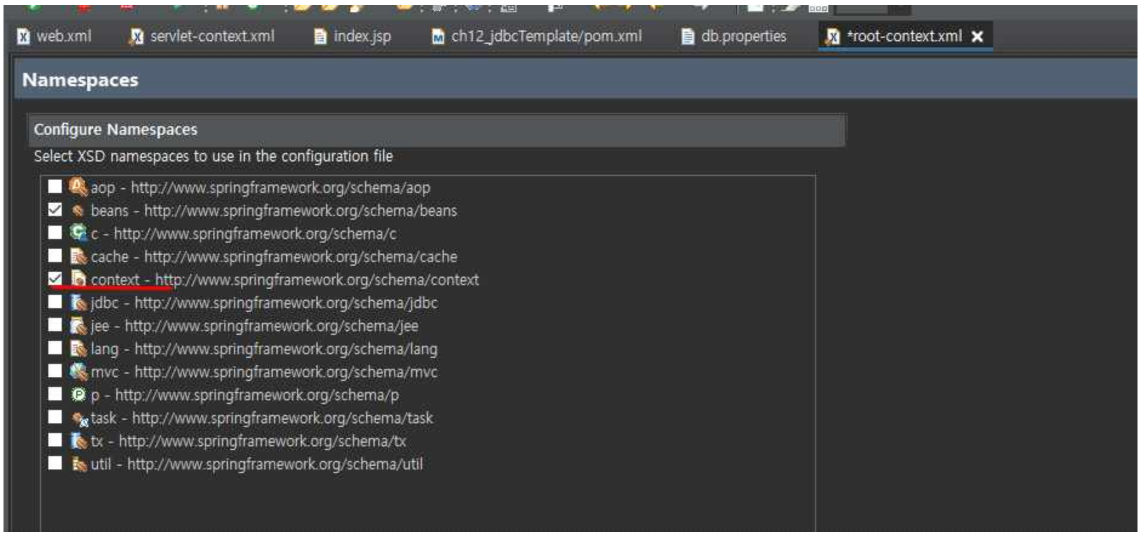
```
db.driverClassName=oracle.jdbc.driver.OracleDriver
db.url=jdbc:oracle:thin:@localhost:1521:xe
db.username=scott
db.password=tiger
```

-db 연결정보 ! 띄어쓰기 금지!

db.driverClassName=oracle.jdbc.driver.OracleDriver

db.url=jdbc:oracle:thin:@localhost:1521:xe
db.username=scott
db.password=tiger 추가!

-root 에 context 체크



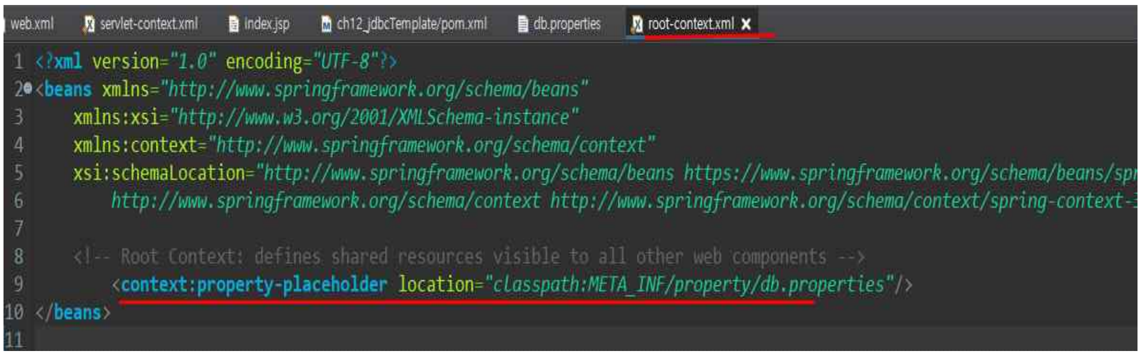
Namespaces

Configure Namespaces

Select XSD namespaces to use in the configuration file

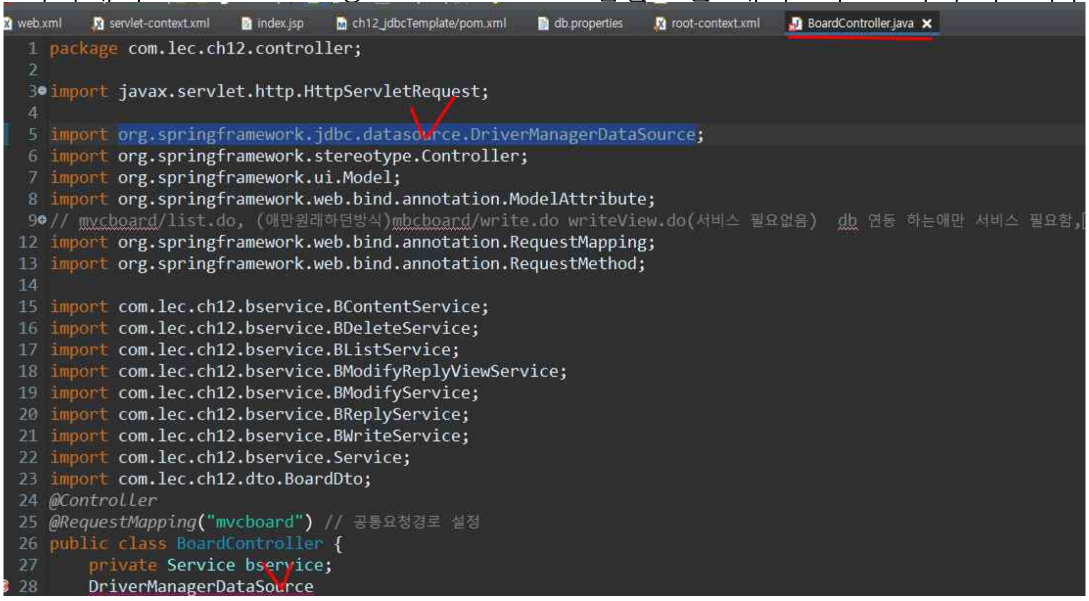
- [ ] aop - http://www.springframework.org/schema/aop
- [✓] beans - http://www.springframework.org/schema/beans
- [ ] c - http://www.springframework.org/schema/c
- [ ] cache - http://www.springframework.org/schema/cache
- [✓] context - http://www.springframework.org/schema/context
- [ ] jdbc - http://www.springframework.org/schema/jdbc
- [ ] jee - http://www.springframework.org/schema/jee
- [ ] lang - http://www.springframework.org/schema/lang
- [ ] mvc - http://www.springframework.org/schema/mvc
- [ ] p - http://www.springframework.org/schema/p
- [ ] task - http://www.springframework.org/schema/task
- [ ] tx - http://www.springframework.org/schema/tx
- [ ] util - http://www.springframework.org/schema/util

- 체크후 추가



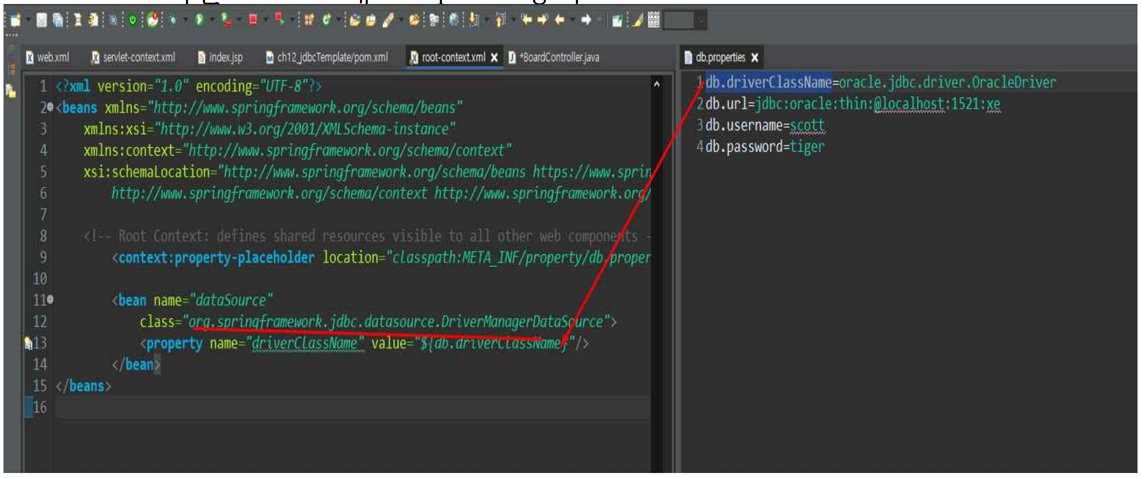
```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/beans https://www.springframework.org/schema/beans/spr
        http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-

    <!-- Root Context: defines shared resources visible to all other web components -->
        <context:property-placeholder location="classpath:META_INF/property/db.properties"/>
</beans>
```

-자바에서 DriverManagerDataSource 임폴트만 해서 주소 복사 후 지우기



```java
package com.lec.ch12.controller;

import javax.servlet.http.HttpServletRequest;

import org.springframework.jdbc.datasource.DriverManagerDataSource;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.ModelAttribute;
// mvcboard/list.do, (애만원래하던방식)mbcboard/write.do writeView.do(서비스 필요없음)  db 연동 하는애만 서비스 필요함,[
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RequestMethod;

import com.lec.ch12.bservice.BContentService;
import com.lec.ch12.bservice.BDeleteService;
import com.lec.ch12.bservice.BListService;
import com.lec.ch12.bservice.BModifyReplyViewService;
import com.lec.ch12.bservice.BModifyService;
import com.lec.ch12.bservice.BReplyService;
import com.lec.ch12.bservice.BWriteService;
import com.lec.ch12.bservice.Service;
import com.lec.ch12.dto.BoardDto;
@Controller
@RequestMapping("mvcboard") // 공통요청경로 설정
public class BoardController {
    private Service bservice;
    DriverManagerDataSource
```

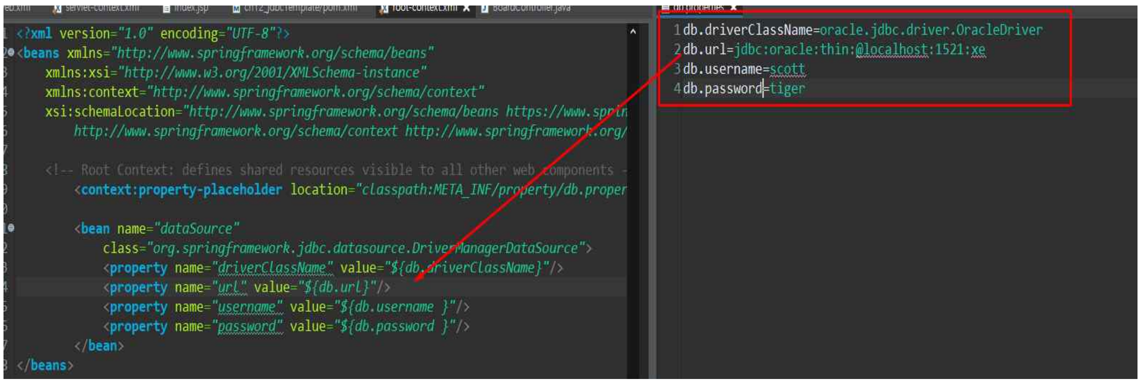
-class = 복붙 value 에 db.pro 넣기



```xml
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/beans https://www.sprin
        http://www.springframework.org/schema/context http://www.springframework.org/

    <!-- Root Context: defines shared resources visible to all other web components -
        <context:property-placeholder location="classpath:META_INF/property/db.proper

        <bean name="dataSource"
            class="org.springframework.jdbc.datasource.DriverManagerDataSource">
            <property name="driverClassName" value="${db.driverClassName}"/>
        </bean>
</beans>
```

```
db.driverClassName=oracle.jdbc.driver.OracleDriver
db.url=jdbc:oracle:thin:@localhost:1521:xe
db.username=scott
db.password=tiger
```

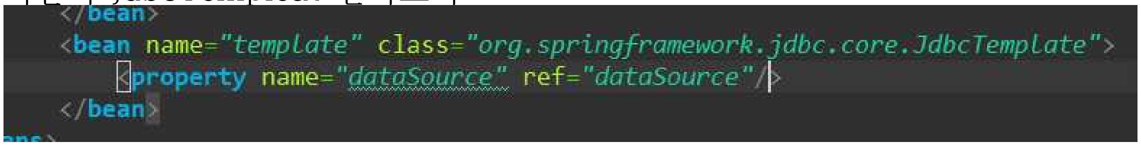
db.pro 받아오기 root-context.xml에 받아오기 !



```
<?xml version="1.0" encoding="UTF-8"?>
<beans xmlns="http://www.springframework.org/schema/beans"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:context="http://www.springframework.org/schema/context"
    xsi:schemaLocation="http://www.springframework.org/schema/beans https://www.sprin
        http://www.springframework.org/schema/context http://www.springframework.org/

    <!-- Root Context: defines shared resources visible to all other web components -
        <context:property-placeholder location="classpath:META_INF/property/db.proper

        <bean name="dataSource"
            class="org.springframework.jdbc.datasource.DriverManagerDataSource">
            <property name="driverClassName" value="${db.driverClassName}"/>
            <property name="url" value="${db.url}"/>
            <property name="username" value="${db.username }"/>
            <property name="password" value="${db.password }"/>
        </bean>
</beans>
```

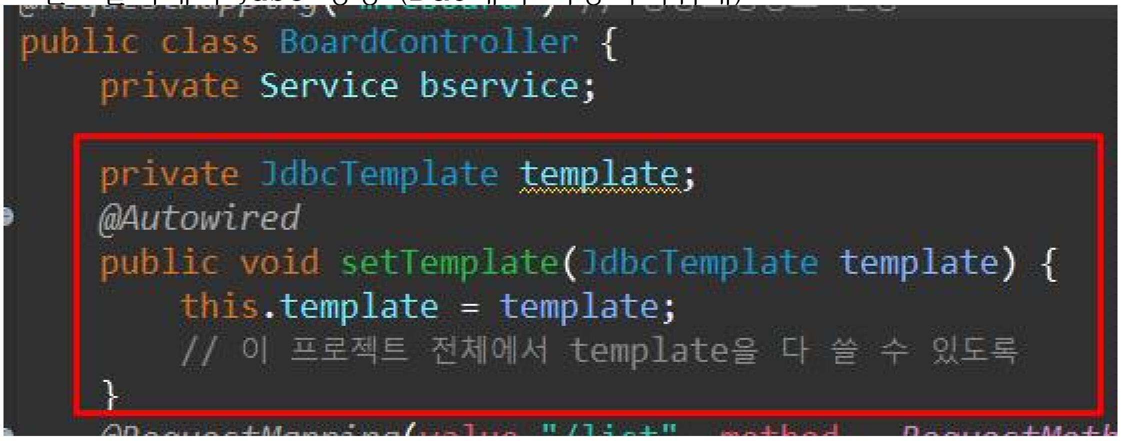
```
db.driverClassName=oracle.jdbc.driver.OracleDriver
db.url=jdbc:oracle:thin:@localhost:1521:xe
db.username=scott
db.password=tiger
```

똑같이 jdbcTempleat 받아오기



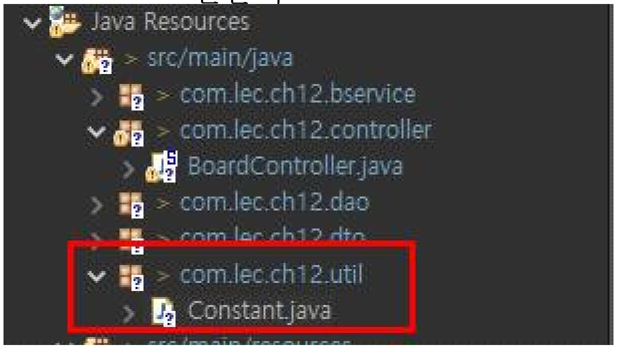
```
    </bean>
    <bean name="template" class="org.springframework.jdbc.core.JdbcTemplate">
        <property name="dataSource" ref="dataSource"/>
    </bean>
```

- 컨트롤러에서 jdbc 생성 (Dao에서 사용하기위해)



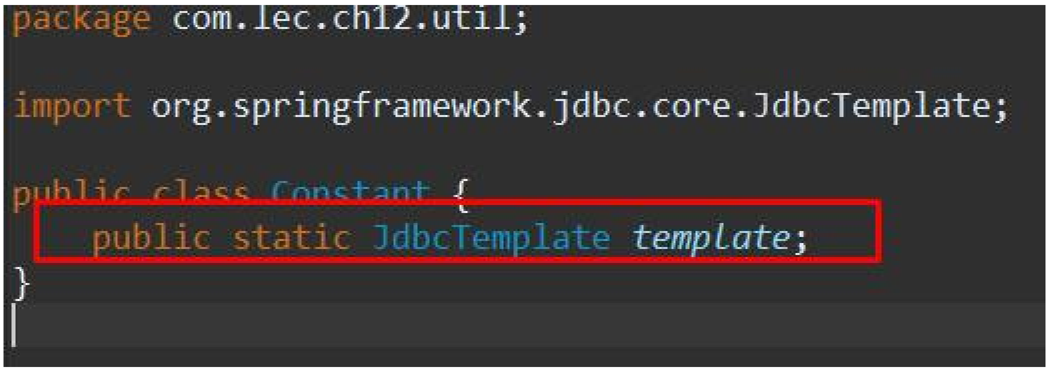
```
public class BoardController {
    private Service bservice;

    private JdbcTemplate template;
    @Autowired
    public void setTemplate(JdbcTemplate template) {
        this.template = template;
        // 이 프로젝트 전체에서 template을 다 쓸 수 있도록
    }
```

- Constant 만들기



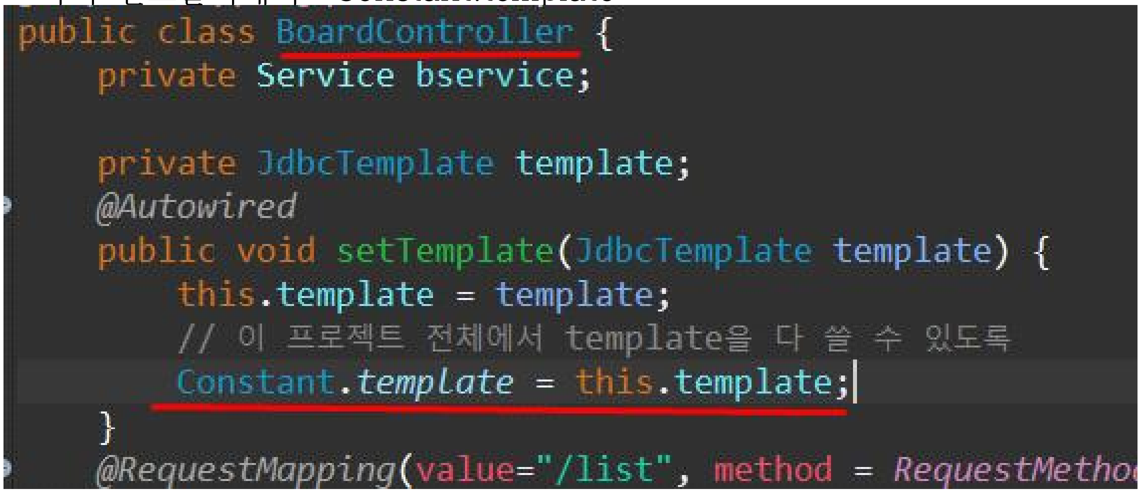
```
Java Resources
  src/main/java
    com.lec.ch12.bservice
    com.lec.ch12.controller
      BoardController.java
    com.lec.ch12.dao
    com.lec.ch12.dto
    com.lec.ch12.util
      Constant.java
```

- Constant Jdbc 만들기



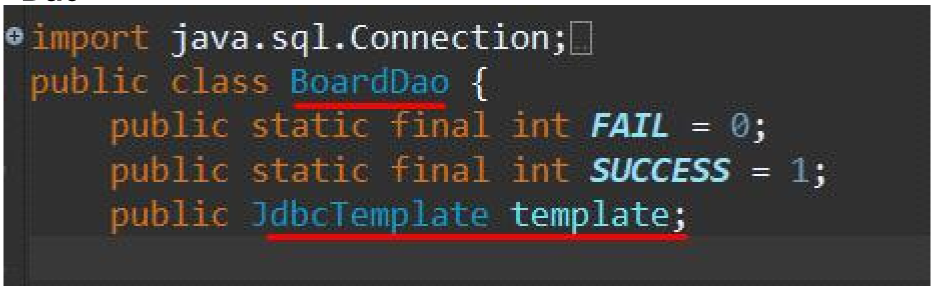
```
package com.lec.ch12.util;

import org.springframework.jdbc.core.JdbcTemplate;

public class Constant {
    public static JdbcTemplate template;
}
```

- 다시 컨트롤러에서 Constant.template

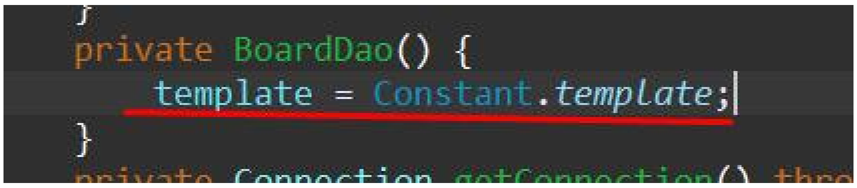


```
public class BoardController {
    private Service bservice;

    private JdbcTemplate template;
    @Autowired
    public void setTemplate(JdbcTemplate template) {
        this.template = template;
        // 이 프로젝트 전체에서 template을 다 쓸 수 있도록
        Constant.template = this.template;
    }
    @RequestMapping(value="/list", method = RequestMetho
```

-Dao

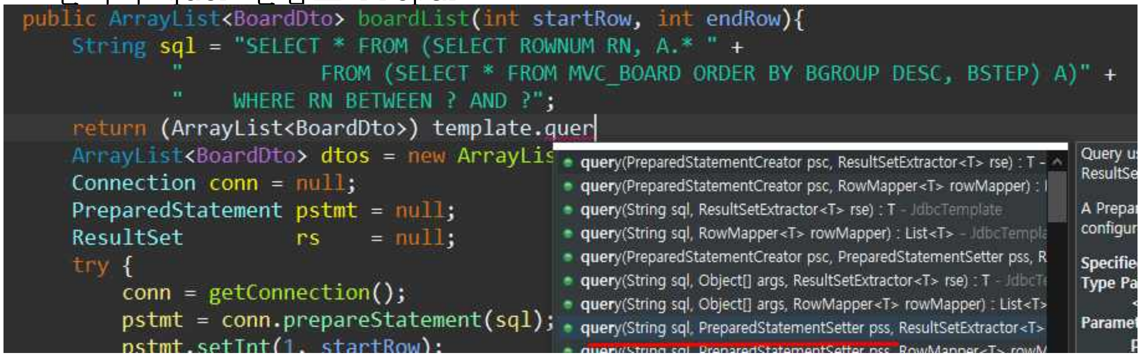


```
import java.sql.Connection;
public class BoardDao {
    public static final int FAIL = 0;
    public static final int SUCCESS = 1;
    public JdbcTemplate template;
```



```
    private BoardDao() {
        template = Constant.template;
    }
```

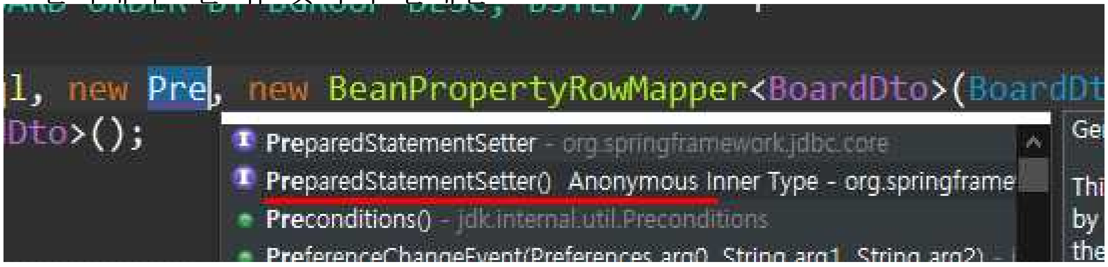
- 글목록 quer 물음표 Prepar



```
public ArrayList<BoardDto> boardList(int startRow, int endRow){
    String sql = "SELECT * FROM (SELECT ROWNUM RN, A.* " +
            "           FROM (SELECT * FROM MVC_BOARD ORDER BY BGROUP DESC, BSTEP) A)" +
            "    WHERE RN BETWEEN ? AND ?";
    return (ArrayList<BoardDto>) template.quer
    ArrayList<BoardDto> dtos = new ArrayLis
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet         rs    = null;
    try {
        conn = getConnection();
        pstmt = conn.prepareStatement(sql);
        pstmt.setInt(1, startRow);
```

```
query(PreparedStatementCreator psc, ResultSetExtractor<T> rse) : T -
query(PreparedStatementCreator psc, RowMapper<T> rowMapper) :
query(String sql, ResultSetExtractor<T> rse) : T - JdbcTemplate
query(String sql, RowMapper<T> rowMapper) : List<T> - JdbcTempla
query(PreparedStatementCreator psc, PreparedStatementSetter pss, R
query(String sql, Object[] args, ResultSetExtractor<T> rse) : T - JdbcTe
query(String sql, Object[] args, RowMapper<T> rowMapper) : List<T>
query(String sql, PreparedStatementSetter pss, ResultSetExtractor<T>
```

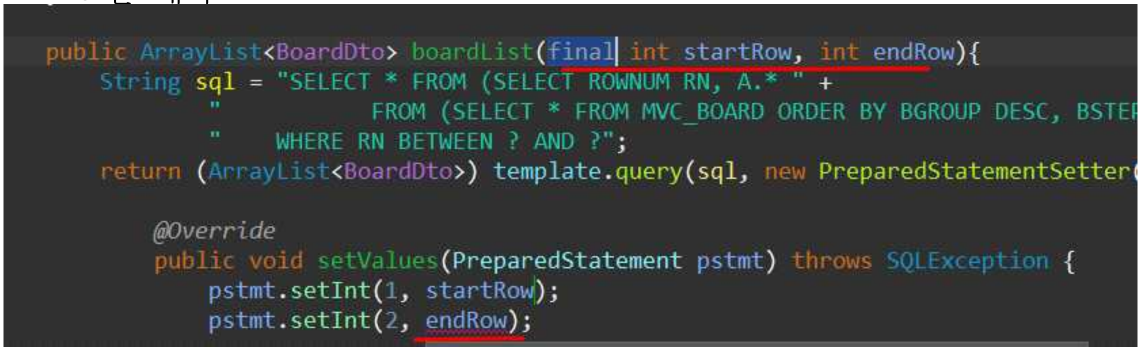
- 인터페이스는 뉴 못하니 밑에껄로



```
l, new Pre, new BeanPropertyRowMapper<BoardDto>(BoardDt
Dto>();
```

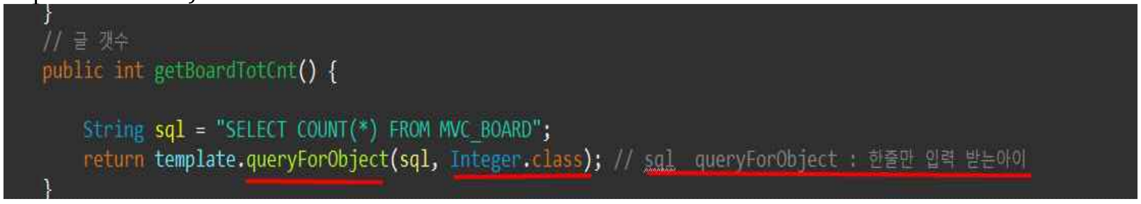
```
PreparedStatementSetter - org.springframework.jdbc.core
PreparedStatementSetter() Anonymous Inner Type - org.springframe
Preconditions() - jdk.internal.util.Preconditions
```

- final을 해라



```
public ArrayList<BoardDto> boardList(final int startRow, int endRow){
    String sql = "SELECT * FROM (SELECT ROWNUM RN, A.* " +
            "           FROM (SELECT * FROM MVC_BOARD ORDER BY BGROUP DESC, BSTEP
            "    WHERE RN BETWEEN ? AND ?";
    return (ArrayList<BoardDto>) template.query(sql, new PreparedStatementSetter(

        @Override
        public void setValues(PreparedStatement pstmt) throws SQLException {
            pstmt.setInt(1, startRow);
            pstmt.setInt(2, endRow);
```

-querForObject



```
}
// 글 갯수
public int getBoardTotCnt() {

    String sql = "SELECT COUNT(*) FROM MVC_BOARD";
    return template.queryForObject(sql, Integer.class); // sql  queryForObject : 한줄만 입력 받는아이
}
```