1. pro04/pom.xml 의존성 라이브러리 추가


```xml
154         </dependency> -->
155
156         <!-- 스프링 트랜잭션 라이브러리 -->
157         <dependency>
158             <groupId>org.springframework</groupId>
159             <artifactId>spring-tx</artifactId>
160             <version>${org.springframework-version}</version>
161         </dependency>
162
```
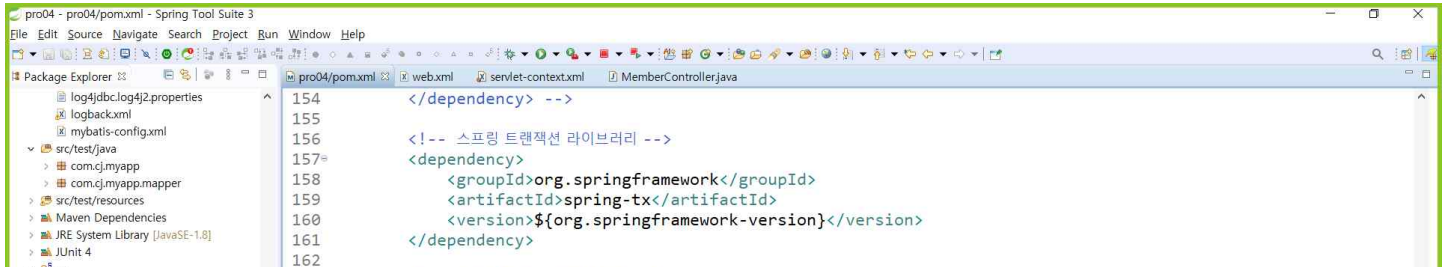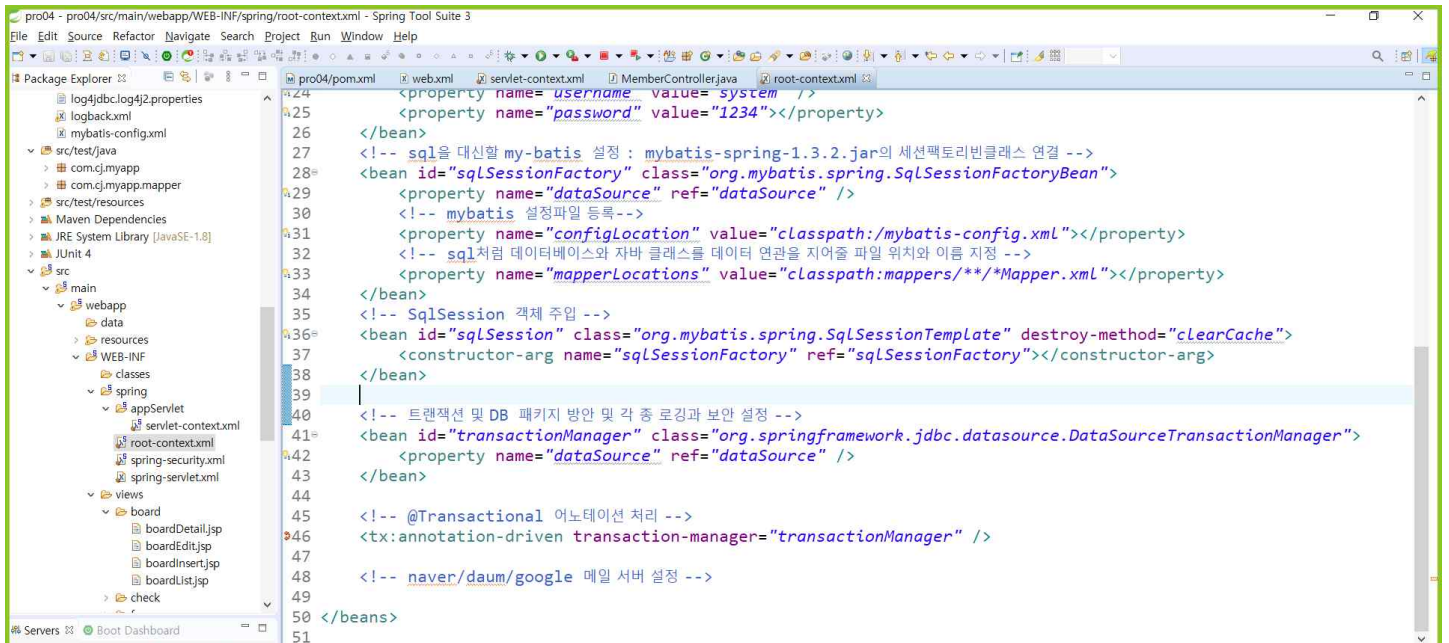
2. pro04/src/main/webapp/WEB-INF/spring/root-context.xml 에 트랜잭션 객체 주입 설정


```xml
24      <property name="username" value="system" />
25      <property name="password" value="1234"></property>
26  </bean>
27  <!-- sql을 대신할 my-batis 설정 : mybatis-spring-1.3.2.jar의 세션팩토리빈클래스 연결 -->
28  <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
29      <property name="dataSource" ref="dataSource" />
30      <!-- mybatis 설정파일 등록-->
31      <property name="configLocation" value="classpath:/mybatis-config.xml"></property>
32      <!-- sql처럼 데이터베이스와 자바 클래스를 데이터 연관을 지어줄 파일 위치와 이름 지정 -->
33      <property name="mapperLocations" value="classpath:mappers/**/*Mapper.xml"></property>
34  </bean>
35  <!-- SqlSession 객체 주입 -->
36  <bean id="sqlSession" class="org.mybatis.spring.SqlSessionTemplate" destroy-method="clearCache">
37      <constructor-arg name="sqlSessionFactory" ref="sqlSessionFactory"></constructor-arg>
38  </bean>
39
40  <!-- 트랜잭션 및 DB 패키지 방안 및 각 종 로깅과 보안 설정 -->
41  <bean id="transactionManager" class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
42      <property name="dataSource" ref="dataSource" />
43  </bean>
44
45  <!-- @Transactional 어노테이션 처리 -->
46  <tx:annotation-driven transaction-manager="transactionManager" />
47
48  <!-- naver/daum/google 메일 서버 설정 -->
49
50  </beans>
51
```

3. pro04/src/main/java/com.cj.dao.FreeDAOImpl.java 에 글을 읽고, 글을 읽었을 때 읽은 횟수를 증가시키도록 @Transactional 애노테이션을 선언하고, 여러 sqlSession 명령을 묶는다.


```java
14
15      @Autowired
16      private SqlSession sqlSession;
17
18      @Override
19      public List<FreeDTO> freeList() throws Exception {
20          return sqlSession.selectList("free.freeList");
21      }
22
23      @Transactional
24      @Override
25      public FreeDTO freeDetail(int bno) throws Exception {
26          sqlSession.update("free.visitCount", bno);
27          return sqlSession.selectOne("free.freeDetail", bno);
28      }
29
30      @Override
31      public void freeInsert(FreeDTO dto) throws Exception {
32          sqlSession.insert("free.freeInsert", dto);
33      }
34
35      @Override
36      public void freeDelete(int bno) throws Exception {
37          sqlSession.delete("free.freeDelete", bno);
38      }
39
```