Case) SqlSessionTemplate sqlSession[1]

1. dao.BoardDaoImpl.java

```
// @Repository은  클래스가 데이터 액세스(데이터 저장소) 계층의 컴포넌트(Bean) 임을 선언한다.
@Repository
// BoardDao 를 상속받는(implements) BoardDaoImpl 함수는
public class BoardDaoImpl implements BoardDao {
// namespace는 앞선 case에서 살펴보았다.
  private final String NAME_SPACE = "com.springstudy.bbs.mapper.BoardMapper";

// SqlSessionTemplate은 마이바티스 스프링 연동모듈의 핵심이다.
  강의때 설명들었지만 막연하게 연동된다고만 기억하고 진행했다.
  private SqlSessionTemplate sqlSession;
  @Autowired
    public void setSqlSession(SqlSessionTemplate sqlSession) {
      this.sqlSession = sqlSession;
    }
```

2. 문제인식**/**정의

private SqlSessionTemplate sqlSession; 이를 보다 상세하게 살펴보고자 한다.
root-context에서 이와 관련된 설정을 진행한 바 있다.

3. 접근**/**해결

https://mybatis.org/spring/ko/sqlsession.html

스프링 Bean 설정 파일인 root-context.xml 파일에 SqlSessionFactory 생성을 위한 SqlSessionFactoryBean을 Spring Bean으로 정의를 해야한다. SqlSessionFactory 객체는 MyBatis와 스프링프레임워크 연동에서 핵심적인 객체로 MyBatis의 전반적인 정보를 가지고 있는 객체이다. 이 객체는 DB Connection을 생성하고 관리하며 SQL 실행에 대한 모든 것을 처리하는 객체로 SqlSessionFactoryBean을 통해 SqlSessionFactory 객체를 한 번만 생성해 사용한다.

```
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
```

스프링 Bean 설정 파일인 root-context.xml 파일에 DAO에서 의존하는 SqlSessionTemplate을 Spring Bean으로 정의를 해야한다. SqlSessionTemplate은 SqlSessionFactoryBean을 사용해 DB 접속해 작업하기 때문에 아래와 같이 생성자로 주입 받도록 설정하면 된다.

```
<bean id="sqlSessionTemplate" class="org.mybatis.spring.SqlSessionTemplate"
    c:sqlSessionFactory-ref="sqlSessionFactory" />
```

추가로, 마이바티스는 JDBC 기반이기 때문에 DataSourceTransactionManager를 이용해 다음과 같은 방식의 트랜잭션을 처리할 수 있다.

```
<bean id="transactionManager"
    class="org.springframework.jdbc.datasource.DataSourceTransactionManager"
    p:dataSource-ref="dataSource" />
```

[1] 이 케이스는 여전히 쉽지 않다. 반복숙달하면서 살펴봐야 겠다.