

## 4. Mybatis

1. dddd

## ■ sql-map-config.xml

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN" "http://mybatis.org/dtd/mybatis-3-config.dtd">
<configuration>
<settings>
  <setting name="jdbcTypeForNull" value="VARCHAR"/>
  <setting name="mapUnderscoreToCamelCase" value="true"/>
</settings>
</configuration>
```

```
<!-- Alias 설정 -->
<typeAliases>
  <typeAlias alias="board" type="com.springbook.biz.board.BoardVO"/>
</typeAliases>

<typeHandlers>
  <!-- java.sql.Timestamp 를 java.util.Date 형으로 반환 -->
  <typeHandler javaType="java.sql.Date" handler="org.apache.ibatis.type.DateTypeHandler"/>
</typeHandlers>
```

```
EmpMapper.java
1 package com.dbal.app.emp.mapper;
2
3 import java.util.List;
4
5
6 public interface EmpMapper {
7     public EmpVO getEmp(EmpVO empVO);
8     public List<EmpVO> getEmpList();
9     public void empInsert(EmpVO empVO);
10 }

```

```
emp_mapper.xml
1 <!DOCTYPE mapper SYSTEM "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
2
3 <mapper namespace="com.dbal.app.emp.mapper.EmpMapper">
4
5     <select id="getEmp"
6             resultType="com.dbal.app.emp.map.EmpVO"
7             parameterType="com.dbal.app.emp.map.EmpVO">
8         select *
9         from employees
10        where employees_id = #{employees_id}
11    </select>
12
13
14 <select id="getEmpList" resultType="com.dbal.app.emp.map.EmpVO">
```

## DAO 클래스 작성

```
package com.dbal.app.emp.mapper;
import java.util.List;
import org.mybatis.spring.SqlSessionTemplate;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Repository;
@Repository
public class EmpDAO {
    @Autowired private SqlSessionTemplate mybatis;

    public EmpVO getEmp(EmpVO empVO) {
        return mybatis.selectOne("com.dbal.app.emp.map.EmpMapper.getEmp",empVO );
    }

    public List<EmpVO> getEmpList() {
        return mybatis.selectList("com.dbal.app.emp.map.EmpMapper.getEmpList");
    }
    public void empInsert(EmpVO empVO) {
        mybatis.insert("com.dbal.app.emp.map.EmpMapper.empInsert", empVO);
    }
}
```

## 트랜잭션

```
<!-- TransactionManager bean 등록 -->
<bean id="txManager"
      class="org.springframework.jdbc.datasource.DataSourceTransactionManager">
  <property name="dataSource" ref="dataSource" />
</bean>
<!-- @Transactional 어노테이션 처리 -->
<tx:annotation-driven transaction-manager="transactionManager" />
```

```
public void insert(EmpVO vo) {  
    try {  
        //1. connect  
        conn = ds.getConnection();  
        //트랜잭션 범위 시작  
        conn.setAutoCommit(false);  
        //2. statement  
        String sql = "INSERT INTO EMPLOYEES~~~ ";  
        PreparedStatement pstmt = conn.prepareStatement(sql);  
        pstmt.executeUpdate();  
        sql = "INSERT INTO MEMBER ~~~ ";  
        pstmt = conn.prepareStatement(sql);  
        int r = pstmt.executeUpdate();  
        //커밋 : 트랜잭션 범위 종료  
        conn.commit();  
  
    } catch (Exception e){  
        if(conn != null)  
            //롤백 : 트랜잭션 범위 종료  
            try { conn.rollback(); } catch (SQLException e1) { }  
    } finally {  
        if(conn != null)  
            try { conn.close(); } catch (SQLException e1) { }  
    }  
}
```

## 프로시저

```

<parameterMap type="board" id="boardParam">
  <parameter property="title" mode="IN" jdbcType="VARCHAR" javaType="string"/>
  <parameter property="writer" mode="IN" jdbcType="VARCHAR" javaType="string"/>
  <parameter property="content" mode="IN" jdbcType="VARCHAR" javaType="string"/>
  <parameter property="seq" mode="OUT" jdbcType="NUMERIC" javaType="int"/>
  <parameter property="out_msg" mode="OUT" jdbcType="VARCHAR" javaType="string"/>
</parameterMap>

<insert id="insertBoardProc1" statementType="CALLABLE" parameterMap="boardParam">
  { call BOARD_INS_PROC(?,?,?,?) }
</insert>

<insert id="insertBoardProc2" statementType="CALLABLE" parameterType="board">
  { call BOARD_INS_PROC(
    #{title},
    #{writer},
    #{content, mode=IN, jdbcType=VARCHAR, javaType=string},
    #{seq, mode=OUT, jdbcType=NUMERIC, javaType=java.math.BigDecimal},
    #{out_msg, mode=OUT, jdbcType=VARCHAR, javaType=string}
  )
  }
</insert>

```