

Mybatis (SQL Mapper)



기존 JDBC 연동 과정은 복잡하고 SQL문이 코드내에 섞여 코드가 상당히 복잡 하다는 단점이 있다. 마이바티스(Mybatis)란 개발자들의 편의성을 위한 스프링에서 데이터베이스 연동을 도와주는 프레임워크 이다.

Mybatis (SQL Mapper)

MyBatis : SQL문을 이용해서 DB에 접근하여, 데이터를 객체화 시켜주고, SQL을 직접 작성하여 수행 결과를 객체와 Mapping 한다. 우리나라의 전자정부 표준프레임워크에서 Spring Mabatis를 채택하고 있기 때문에, 우리나라 공공 프로젝트에서는 Mybatis를 많이 사용하고 있다.

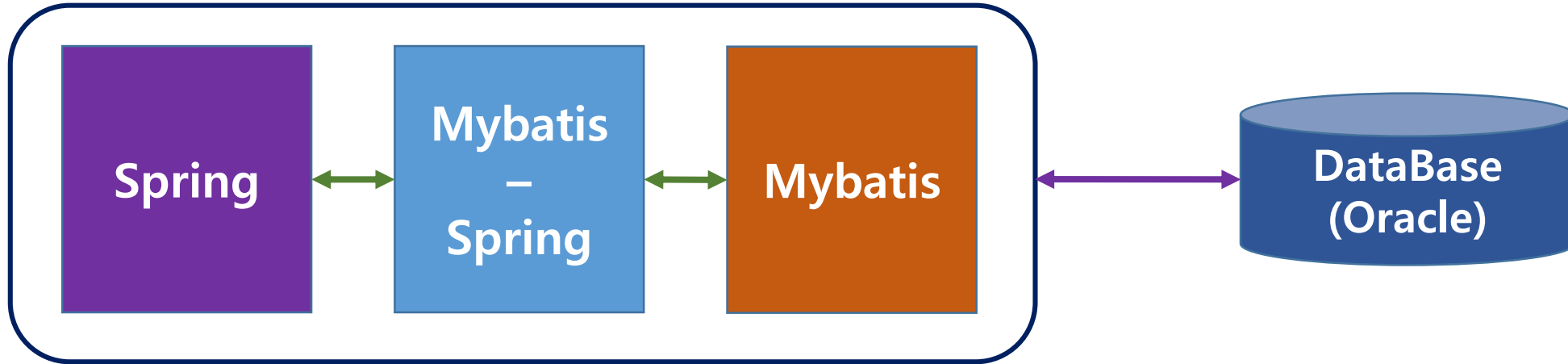
MyBatis 장점

- SQL 문을 xml 로 분리 가능하다.
- 복잡한 SQL 문 작성이 가능하다.

MyBatis 단점

- 객체와 SQL 문 모두 관리해야 한다.
- CRUD 코드를 직접 다 구현해야 한다.

Spring - Mybatis 설정



Mybatis 설정을 위한 필요 라이브러리
Oracle JDBC (ojdbc)
mybatis
mybatis-spring
spring-jdbc



Spring - Mybatis 설정

```
<!-- mybatis -->
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis</artifactId>
  <version>3.5.10</version>
</dependency>
```

```
<!-- mybatis-spring -->
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis-spring</artifactId>
  <version>2.0.6</version>
</dependency>
```

pom.xml 에 <dependency> 태그로 MyBatis 라이브러리를
다운받아 Mybatis 를 사용하기 위한 설정을 해준다.

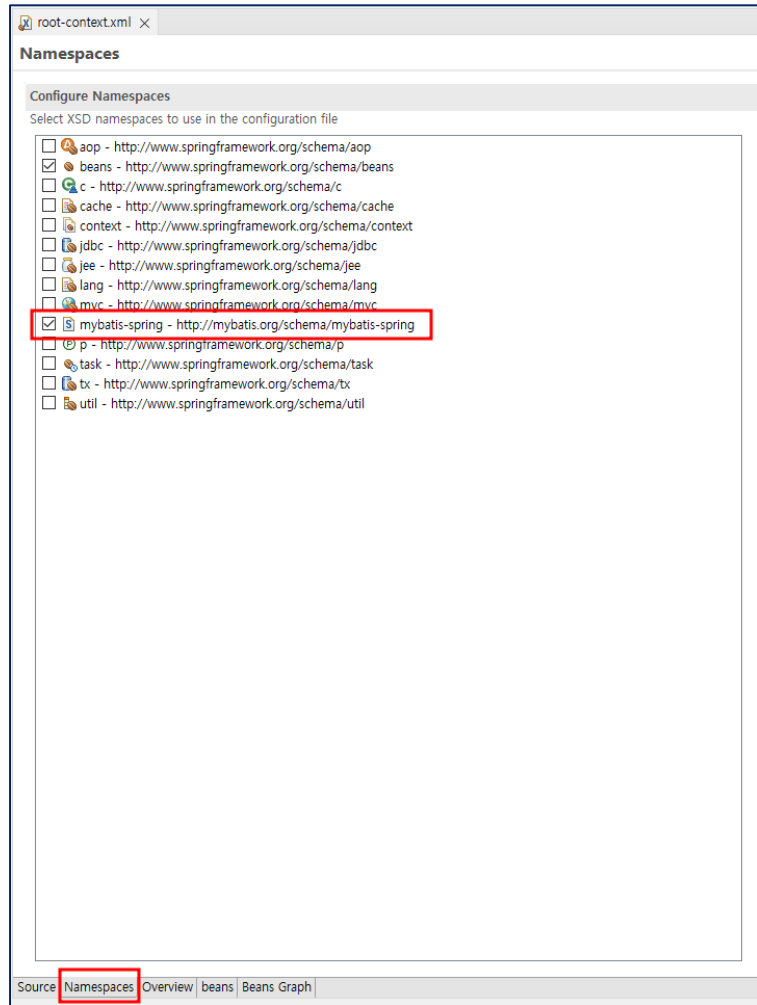


Spring - Mybatis 설정

```
<bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
    <property name="dataSource" ref="ds" />
    <property name="mapperLocations" value="classpath:/mappers/**/*.xml" />
</bean>
```

마이바티스(Mybatis)를 사용하기 위하여 SqlSessionFactoryBean 클래스를 root-context.xml 파일에서 빈으로 등록한다. 또한 DataSource 객체를 참조하여 동작하는 속성 설정과 [**.xml] 파일이 스프링이 동작할 때 같이 동작하도록 mapperLocations 속성을 추가한다.

Spring - Mybatis 설정



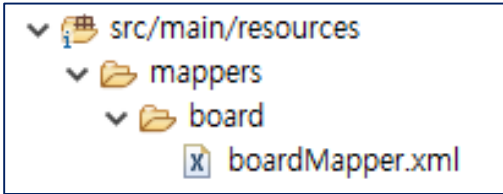
com.spring.web.board.mapper
BoardMapper.java

```
<mybatis-spring:scan base-package="com.spring.web.board.mapper"/>  
</beans>
```

mapper 인터페이스를 넣을 패키지를 만들어준다.
그 안에 인터페이스를 생성하고 root-context.xml
에 추가적인 설정을 하여 스프링이 해당 패키지안
의 인터페이스를 mapper 로 인식하도록 한다.



Spring - Mybatis 설정



XML 저장 위치 : 'src/main/resources' 에 Mapper 가 저장될 패키지명을 폴더 경로로 만들어준다. 해당 위치에 xml 파일을 만든 후 아래 코드를 작성한다.

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
    "https://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="com.spring.web.board.mapper.BoardMapper">

</mapper>
```

마이바티스 Mybatis

SQL 명령어에 따라 SELECT문은 <select>, INSERT문은 <insert>, UPDATE문은 <update>, DELETE문은 <delete>에 작성한다.

```
<select id="boardList" resultMap="boardMap" >
    SELECT * FROM board ORDER BY id DESC
</select>

<insert id="write">
    INSERT INTO board
        (id, writer, title, content)
        VALUES(board_seq.nextval, #{writer}, #{title}, #{content})
</insert>

<update id="modify">
    UPDATE board
        SET title = #{title}, content = #{content}
        WHERE id = #{id}
</update>

<delete id="delete">
    DELETE FROM board
        WHERE id = #{id}
</delete>
```

SQL문 성공 시 결과 값	
select	검색 결과
insert	1
update	성공한 행 개수
delete	

마이바티스 Mybatis

SQL 명령어 태그안에 들어가는 속성

속성	설명
id	인터페이스에 선언된 메소드명을 지정한다. (각 SQL문을 구분)
resultType	SELECT문 실행 결과를 담은 객체를 지정한다. (패키지 이름을 포함한 클래스 이름 또는 객체 alias 지정)
resultMap	SELECT문 실행 결과를 담은 객체를 resultMap으로 지정한다. <resultMap>을 따로 선언해줘야 한다. (resultType과 resultMap 중 하나를 택해서 설정한다.)
parameterType	이 속성에 지정한 객체의 프로퍼티값이 SQL문의 입력 파라미터에 지정된다.

마이바티스 Mybatis

ResultMap이란? SELECT문 실행 결과 컬럼데이터를 저장할 객체의 프로퍼티에 지정함으로써 저장할 객체에 간편하게 담아주기 위한 태그이다.

```
<resultMap id="boardMap" type="com.spring.web.board.dto.BoardDTO" >
    <id property="id" column="id" jdbcType="NUMBER" javaType="Integer" /> <!-- pk 매핑 -->
    <result property="writer" column="writer" jdbcType="NVARCHAR" javaType="String" />
    <result property="title" column="title" jdbcType="NVARCHAR" javaType="String" />
    <result property="content" column="content" jdbcType="NVARCHAR" javaType="String" />
    <result property="regDate" column="reg_date" jdbcType="TIMESTAMP" javaType="Timestamp" />
    <result property="viewCount" column="view_count" jdbcType="NUMBER" javaType="Integer" />
</resultMap>
```

```
<select id="boardList" resultMap="boardMap" >
    SELECT * FROM board ORDER BY id DESC
</select>
```