

Spring Framework

17. MyBatis 어플리케이션 작성(1)

CONTENTS

1

MyBatis와 MyBatis-Spring 설치

2

Mapping 파일 작성 및 MyBatis 설정

3

DAO 클래스 작성 및 테스트

학습 목표

- MyBatis 및 MyBatis-Spring 설치에 대하여 이해할 수 있습니다.
- Mapping 파일 작성 및 MyBatis 설정에 대하여 이해할 수 있습니다.
- DAO 클래스 작성 및 테스트에 대하여 이해할 수 있습니다.

A person's hands are shown holding a smartphone, with the screen glowing. The background is dark with out-of-focus, colorful bokeh lights in shades of yellow, orange, and blue. A semi-transparent dark banner is at the bottom, containing a yellow decorative element and the title text.

1. MyBatis와 MyBatis-Spring 설치

■ MyBatis 라이브러리 검색 및 설치

<http://mvnrepository.com>에 접근한다.

[mybatis](#) 로 검색한다.

[mybatis 3.4.1](#) 버전을 [pom.xml](#)에 추가한다.

```
<!-- https://mvnrepository.com/artifact/org.mybatis/mybatis -->
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis</artifactId>
  <version>3.4.1</version>
</dependency>
```

■ MyBatis-Spring 라이브러리 검색 및 설치

<http://mvnrepository.com>에 접근한다.

`mybatis spring`로 검색한다.

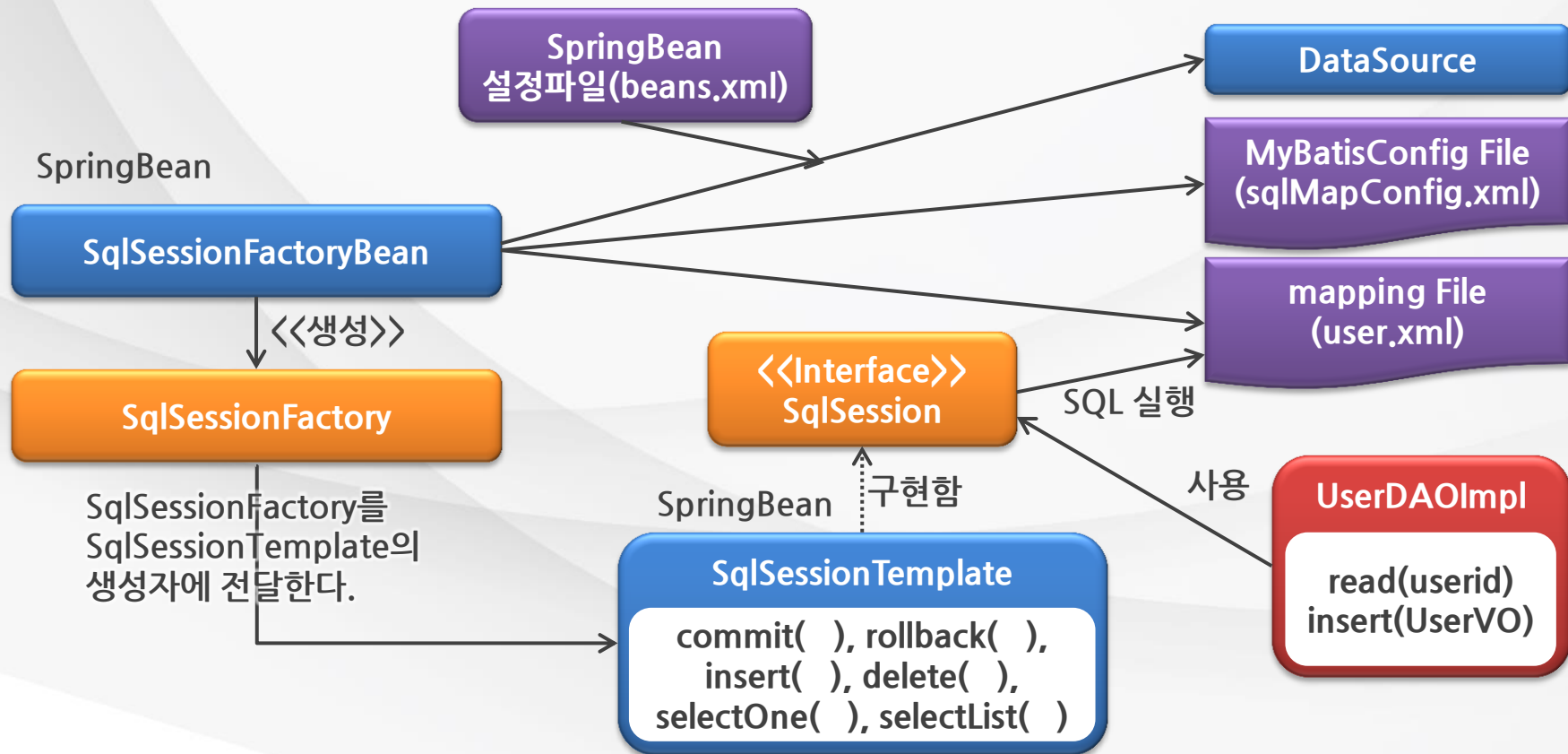
`mybatis spring 1.3.0` 버전을 `pom.xml`에 추가한다.

```
<!-- https://mvnrepository.com/artifact/org.mybatis/mybatis-spring -->
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis-spring</artifactId>
  <version>1.3.0</version>
</dependency>
```

A person's hands are shown holding a black smartphone with a white screen. The background is dark with out-of-focus, colorful bokeh lights in shades of yellow, orange, and blue. A semi-transparent dark banner with a yellow decorative element on the left is positioned at the bottom of the image.

2. Mapping 파일 작성 및 MyBatis 설정

■ MyBatis-Spring의 주요 컴포넌트



■ MyBatis Config 파일 작성(SqlMapConfig.xml)

```
SqlMapConfig.xml
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE configuration PUBLIC "-//mybatis.org//DTD Config 3.0//EN"
3   "http://mybatis.org/dtd/mybatis-3-config.dtd">
4
5 <configuration>
6   <typeAliases>
7     <typeAlias alias="User" type="myspring.user.vo.UserVO" />
8   </typeAliases>
9 </configuration>
```

```
UserVO.java
1 package myspring.user.vo;
2
3 public class UserVO {
4
5   private String userId;
6   private String name;
7   private String gender;
8   private String city;
```

■ 설정파일과 VO 객체와 테이블간의 관계

SqlMapConfig.xml

```
<typeAlias alias="User" type="myspring.user.vo.UserVO" />
```

User.xml

```
<select id="selectUserList" resultType="User">
    select * from users order by userid
</select>
<insert id="insertUser" parameterType="User">
    insert into users values(#{userId},#{name},#{gender},#{city} )
</insert>
```

UserVO.java

```
1 package myspring.user.vo;
2
3 public class UserVO {
4
5     private String userId;
6     private String name;
7     private String gender;
8     private String city;
```

CREATE TABLE USERS

```
(
    userid    VARCHAR2(30) NOT NULL PRIMARY KEY,
    name      VARCHAR2(100) NOT NULL,
    gender    VARCHAR2(10),
    city      VARCHAR2(30)
);
```

■ Mapping 파일 작성(1) : 조회 SQL 문

```
User.xml
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
3   "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
4 <mapper namespace="userNS">
5
6   <select id="selectUserById" parameterType="string" resultType="User">
7     select * from users where userid=#{value}
8   </select>
9
10  <select id="selectUserList" resultType="User">
11    select * from users order by userid
12  </select>
```

■ Mapping 파일 작성(2) : 등록, 수정, 삭제 SQL 문

```
*User.xml
14 <insert id="insertUser" parameterType="User">
15     insert into users values(#{userId},#{name},#{gender},#{city} )
16 </insert>
17
18 <update id="updateUser" parameterType="User">
19     update users set
20     name =
21     #{name},
22     gender = #{gender},
23     city = #{city}
24     where userid = #{userId}
25 </update>
26
27 <delete id="deleteUser" parameterType="string">
28     delete from users where
29     userid = #{value}
30 </delete>
31 </mapper>
```


■ Spring Bean 설정 파일

```
beans.xml
9
10 <!-- SqlSessionFactoryBean 설정 -->
11 <bean id="sqlSessionFactory" class="org.mybatis.spring.SqlSessionFactoryBean">
12     <property name="dataSource" ref="dataSource" />
13     <property name="configLocation" value="classpath:config/SqlMapConfig.xml" />
14     <property name="mapperLocations">
15         <list>
16             <value>classpath:config/User.xml</value>
17         </list>
18     </property>
19 </bean>
20 <!-- SqlSessionTemplate 설정 -->
21 <bean id="sqlSession" class="org.mybatis.spring.SqlSessionTemplate">
22     <constructor-arg ref="sqlSessionFactory" />
23 </bean>
```

A person's hands are shown holding a smartphone, with the screen glowing. The background is dark with out-of-focus, colorful bokeh lights in shades of yellow, orange, and blue. A semi-transparent dark banner is at the bottom, containing a yellow decorative element and the section title.

3. DAO 클래스 작성 및 테스트

■ SqlSessionTemplate을 사용한 DAO 클래스 작성(1)



```
UserDaoImpl.java x User.xml
1 package myspring.user.dao;
2
3 import java.util.List;
10 @Repository("userDao")
11 public class UserDaoImpl implements UserDao {
12     @Autowired
13     private SqlSession session;
14
15     @Override
16     public UserVO read(String id) {
17         UserVO user = session.selectOne("userNS.selectUserById", id);
18         return user;
19     }
20
21     public List<UserVO> readAll() {
22         List<UserVO> userList = session.selectList("userNS.selectUserList");
23         return userList;
24     }
}
```

■ SqlSessionTemplate을 사용한 DAO 클래스 작성(2)

```
*UserDaoImpl.java  User.xml

26 public void insert(UserVO user) {
27     session.update("userNS.insertUser", user);
28     System.out.println("등록된 Record UserId=" + user.getUserId() +
29         " Name=" + user.getName());
30 }
31
32 @Override
33 public void update(UserVO user) {
34     session.update("userNS.updateUser", user);
35 }
36
37 @Override
38 public void delete(String id) {
39     session.delete("userNS.deleteUser", id);
40     System.out.println("삭제된 Record with ID = " + id );
41 }
42 }
```


DAO 클래스 테스트 : 사용자 정보 조회

```
*UserController.java ✕
21 public class UserController {
22     @Autowired
23     ApplicationContext context;
24     @Autowired
25     UserService service;
26
27     @Test
28     public void getUserTest() {
29         UserVO user = service.getUser("gildong");
30         System.out.println("User 정보 : " + user);
31         assertEquals("홍길동", user.getName());
32     }
```

실행 결과

java.lang.IllegalStateException: Failed to load ApplicationContext
Annotation-specified bean name 'userDao' for bean class [myspring.user.dao.UserDaoImplJDBC] conflicts with existing, non-compatible bean definition of same name and class [myspring.user.dao.UserDaoImpl]

DAO 클래스 테스트 : 사용자 등록 및 목록 조회

UserClient.java

```
21 public class UserClient {
22     @Autowired
23     ApplicationContext context;
24     @Autowired
25     UserService service;
26
27     @Test
28     public void insertUserTest() {
29         service.insertUser(new UserVO("polar", "연아", "여", "경기"));
30
31         for (UserVO user : service.getUserList()) {
32             System.out.println(user);
33         }
34     }
```

실행 결과

UserService.insertUser(..) 실행 시간 : 1090
UserService.getUserList() 실행 시간 : 143 ms

DAO 클래스 테스트 : 사용자 정보 수정

```
UserClient.java
21 public class UserClient {
22     @Autowired
23     ApplicationContext context;
24     @Autowired
25     UserService service;
26
27     @Test
28     public void updateUserTest() {
29         service.updateUser(new UserVO("polar", "연아2", "여2", "경기2"));
30         System.out.println(service.getUser("polar"));
31     }
```

실행 결과

UserService.updateUser(..) 실행 시간 : 2031 ms

UserService.getUser(..) 실행 시간 : 189 ms

DAO 클래스 테스트 : 사용자 정보 삭제 및 목록조회

```
UserClient.java
21 public class UserClient {
22     @Autowired
23     ApplicationContext context;
24     @Autowired
25     UserService service;
26
27     @Test
28     public void deleteUserTest() {
29         service.deleteUser("polar");
30
31         for (UserVO user : service.getUserList()) {
32             System.out.println(user);
33         }
34     }
```

실행 결과

UserService.deleteUser(..) 실행 시간 : 2066 ms
UserService.getUserList() 실행 시간 : 209 ms



학습정리

지금까지 **[MyBatis 어플리케이션 작성(1)]**에 대해서 살펴보았습니다.

MyBatis 및 MyBatis-Spring 설치

- ◉ MyBatis 설치
- ◉ MyBatis-Spring 설치

Mapping 파일작성 및 MyBatis 설정

- ◉ Mapping 파일 작성
- ◉ MyBatis 설정

DAO 클래스 작성 및 테스트

- ◉ DAO 클래스 작성
- ◉ DAO 클래스 테스트