

1. 프로젝트 만들기

The screenshots show the following steps in the Maven IDE wizard:

- Select a wizard:** The 'Maven' wizard is selected under the 'Wizards' list.
- New Maven Project:** The 'New Maven project' dialog is shown. The 'Use default workspace location' checkbox is checked. The 'Next >' button is highlighted.
- Select an Archetype:** The 'Select an Archetype' dialog is shown. The 'Filter' is set to 'org.apache.maven.archetypes'. The 'maven-archetype-quickstart' archetype is selected.
- Specify Archetype parameters:** The 'Specify Archetype parameters' dialog is shown. The 'Group Id' is 'com', the 'Artifact Id' is 'springmybatisoracle', and the 'Version' is '0.0.1-SNAPSHOT'.
- Project Structure:** The project structure is shown in the IDE. The 'JRE System Library [JavaSE-1.7]' is highlighted.

2. 메이븐 프로젝트 설정파일 편집: pom.xml

The screenshots show the Maven Central repository search results for various dependencies. The following table summarizes the information:

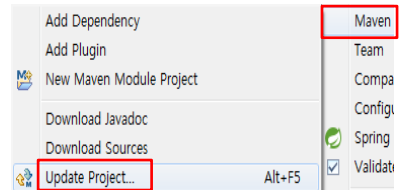
Dependency	Version	Vulnerabilities	Repository	Usages
MvBatis	2020.02.02	1 vulnerability	Central	
Apache Commons DBCP	2019.08.07	2.7.x, 2.7.0	Central	
MyBatis Spring	2020.06.05	2.0.5	Central	
Other	11.2.0.4		Central	68

저장소 사이트: maven central repository
 검색어
 MyBatis, Common DBCP, MyBatis Spring, Ojdbc6

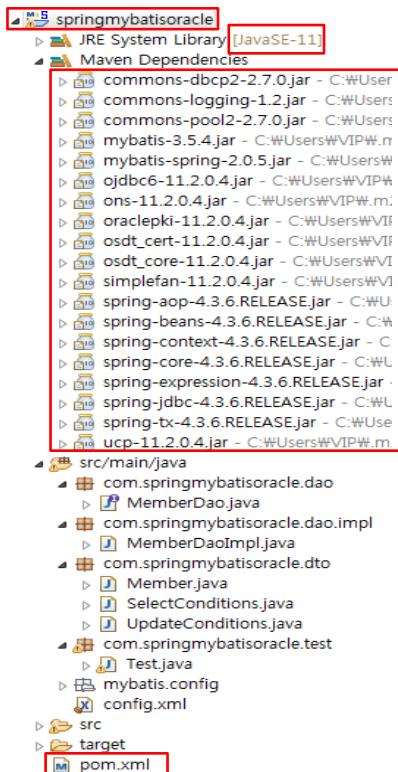
```

springmvcmybatisoracle/pom.xml
27<dependency>
28  <groupId>org.springframework</groupId>
29  <artifactId>spring-jdbc</artifactId>
30  <version>${springframework.version}</version>
31</dependency>
32<dependency>
33  <groupId>org.mybatis</groupId>
34  <artifactId>mybatis</artifactId>
35  <version>3.5.4</version>
36</dependency>
37<dependency>
38  <groupId>org.mybatis</groupId>
39  <artifactId>mybatis-spring</artifactId>
40  <version>2.0.5</version>
41</dependency>
42<dependency>
43  <groupId>org.apache.commons</groupId>
44  <artifactId>commons-dbcp2</artifactId>
45  <version>2.7.0</version>
46</dependency>
47
48<dependency>
49  <groupId>com.oracle.database.jdbc</groupId>
50  <artifactId>ojdbc6</artifactId>
51  <version>11.2.0.4</version>
52</dependency>
53
54<dependency>
55
56</dependency>
57</dependencies>
58<build>
59  <pluginManagement>
60    <plugins>
61      <plugin>
62        <groupId>org.apache.maven.plugins</groupId>
63        <artifactId>maven-compiler-plugin</artifactId>
64        <version>3.8.0</version>
65      </plugin>
66    </plugins>
67  </pluginManagement>
68  <configuration>
69    <source>11</source>
70    <target>11</target>
71  </configuration>
72</build>
73
74</pom>
75
76

```



3. 완성된 기본 프로젝트



4. 테이블 생성(kimht/kimht)

member	
id	varchar2(20)
password	varchar2(20)
name	varchar2(20)
address	varchar2(50)

5. DTO 클래스 만들기

```
config.xml Member.java
1 package com.springmybatis.dto;
2
3 public class Member {
4
5     private String id;
6     private String password;
7     private String name;
8     private String address;
9
10    public Member(String id, String password, String name, String address) {
11        super();
12        this.id = id;
13        this.password = password;
14        this.name = name;
15        this.address = address;
16    }
17
18    public String getId() {
19        return id;
20    }
21
22    public void setId(String id) {
23        this.id = id;
24    }
25
26    public String getPassword() {
27        return password;
28    }
29
30    public void setPassword(String password) {
31        this.password = password;
32    }
33
34    public String getName() {
35        return name;
36    }
37
38    public void setName(String name) {
39        this.name = name;
40    }
41
42    public String getAddress() {
43        return address;
44    }
45
46    public void setAddress(String address) {
47        this.address = address;
48    }
49
50    @Override
51    public String toString() {
52        return "Member [id=" + id + ", password=" + password + ", name=" + name + ", address=" + address + "]"
53    }
54 }
```

6. DAO 클래스 만들기(인터페이스와 구현체 만들기)

MemberDao.java

```
1 package com.springmybatis.dao;
2
3 import com.springmybatis.dto.Member;
4
5 public interface MemberDao {
6
7     public Member findMemberById(String id);
8 }
```

MemberDaoImpl.java

```
1 package com.springmybatis.dao.impl;
2
3 import org.mybatis.spring.SqlSessionTemplate;
4
5 import com.springmybatis.dao.MemberDao;
6 import com.springmybatis.dto.Member;
7
8 public class MemberDaoImpl implements MemberDao {
9
10     private SqlSessionTemplate sqlSessionTemplate;
11
12     public MemberDaoImpl(SqlSessionTemplate sqlSessionTemplate)
13     {
14         super();
15         this.sqlSessionTemplate = sqlSessionTemplate;
16     }
17
18     @Override
19     public Member findMemberById(String id) {
20         return sqlSessionTemplate.selectOne(
21             "member.findMemberById", id);
22     }
23 }
```

주입대상 객체

member.findMemberId

7. 스프링 프레임워크 설정파일 만들기: config.xml

config.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <beans xmlns="http://www.springframework.org/schema/beans"
3       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4       xmlns:context="http://www.springframework.org/schema/context"
5       xmlns:p="http://www.springframework.org/schema/p"
6       xmlns:c="http://www.springframework.org/schema/c"
7       xsi:schemaLocation="http://www.springframework.org/schema/beans
8         http://www.springframework.org/schema/beans/spring-beans.xsd
9         http://www.springframework.org/schema/context
10        http://www.springframework.org/schema/context/spring-context.xsd">
11
12     <bean
13         class="org.apache.commons.dbcp2.BasicDataSource"
14         name="dbcp"
15         p:driverClassName="com.mysql.jdbc.Driver"
16         p:url="jdbc:mysql://localhost/mydb"
17         p:username="kimht" p:password="kimht" />
18
19     <bean
20         class="org.mybatis.spring.SqlSessionFactoryBean"
21         name="sqlSessionFactoryBean"
22         p:dataSource-ref="dbcp"
23         p:mapperLocations="classpath:/mybatis/config/*.xml"
24         p:typeAliasesPackage="com.springmybatis.dto" />
25
26     <bean
27         class="org.mybatis.spring.SqlSessionTemplate"
28         name="sqlSessionTemplate"
29         c:sqlSessionFactory-ref="sqlSessionFactoryBean" />
30
31     <bean class="com.springmybatis.dao.impl.MemberDaoImpl"
32         name="memberDao">
33         <constructor-arg ref="sqlSessionTemplate" />
34     </bean>
35
36 </beans>
```

8. SQL 매퍼 만들기

```
member.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <!-- Sql Mapper -->
3 <!DOCTYPE mapper
4 PUBLIC "-//mybatis.org/DTD Mapper 3.0//EN"
5 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
6 <mapper namespace="member">
7   <select id="findMemberId" resultType="member">
8     SELECT id, password, name, address
9     FROM member
10    WHERE id=#{value}
11  </select>
12 </mapper>
```

9. 데이터 조회 테스트 및 실행

```
Test.java
1 package com.springmybatis.test;
2
3 import org.springframework.context.ApplicationContext;
4
5 public class Test {
6
7     private static ApplicationContext context;
8
9     public static void main(String[] args) {
10         context = new ClassPathXmlApplicationContext("config.xml");
11         MemberDao memberDao =
12             (MemberDao) context.getBean("memberDao");
13         Member member = memberDao.findMemberById("python");
14         System.out.println(member);
15     }
16 }
```

Problems @ Javadoc Declaration Console Coverage Debug

<terminated> Test (23) [Java Application] D:\Java\jdk-11.0.16\bin\javaw.exe (2023. 4. 13: 4월 13, 2023 2:33:30 오후 org.springframework.context.support.ClassPathXmlApplicationContext: Refreshing org.springframework.context.support.ClassPathXmlApplicationContext: 4월 13, 2023 2:33:30 오후 org.springframework.beans.factory.xml.XmlBeanDefinitionReader: Loading XML bean definitions from class path resource [config.xml] Member[id=python, password=python1234, name=python, address=python])

10. 데이터 입력 테스트 및 실행

```
MemberDaoImpl.java member.xml
1 package com.springmybatis.dao.impl;
2
3 import org.mybatis.spring.SqlSessionTemplate;
4
5 public class MemberDaoImpl implements MemberDao {
6
7     private SqlSessionTemplate sqlSessionTemplate;
8
9     public MemberDaoImpl(SqlSessionTemplate sqlSessionTemplate) {
10         super();
11         this.sqlSessionTemplate = sqlSessionTemplate;
12     }
13
14     @Override
15     public Member findMemberById(String id) {
16         return sqlSessionTemplate.selectOne(
17             "member.findMemberId", id);
18     }
19
20     @Override
21     public int insertMember(Member member) {
22         return sqlSessionTemplate.insert(
23             "member.insertMember", member);
24     }
25 }
```

```
MemberDao.java
1 package com.springmybatis.dao;
2
3 import com.springmybatis.dto.Member;
4
5 public interface MemberDao {
6
7     public Member findMemberById(String id);
8
9     public int insertMember(Member member);
10 }
```

```

1 MemberDaoImpl.java member.xml Test.java
2 1 <?xml version="1.0" encoding="UTF-8"?>
3 2 <!-- Sql Mapper -->
4 3 <!DOCTYPE mapper
5 4 PUBLIC "-//mybatis.org/DTD Mapper 3.0//EN"
6 5 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
7 6 <mapper namespace="member"> namespace="member"
8 7 <select id="findMemberId" resultType="member">
9 8 SELECT id, password, name, address
10 9 FROM member
11 10 WHERE id=#{value}
12 11 </select>
13 12
14 13 <insert id="enrollMember" parameterType="com.springmybatis.dto.Member">
15 14 INSERT INTO Member VALUES(#{id}, #{password}, #{name}, #{address})
16 15 </insert>
17 16 </mapper>

```

```

1 MemberDaoImpl.java member.xml
2 package com.springmybatis.dao.impl;
3 import org.mybatis.spring.SqlSessionTemplate;
4
5 public class MemberDaoImpl implements MemberDao {
6
7     private SqlSessionTemplate sqlSessionTemplate;
8
9     public MemberDaoImpl(SqlSessionTemplate sqlSessionTemplate) {
10         super();
11         this.sqlSessionTemplate = sqlSessionTemplate;
12     }
13
14     @Override
15     public Member findMemberById(String id) {
16         return sqlSessionTemplate.selectOne(
17             "member.findMemberId", id);
18     }
19
20     @Override
21     public int insertMember(Member member) {
22         return sqlSessionTemplate.insert("member.", member);
23     }
24 }

```

```

1 MemberDaoImpl.java member.xml Test.java
2 1 <?xml version="1.0" encoding="UTF-8"?>
3 2 <!-- Sql Mapper -->
4 3 <!DOCTYPE mapper
5 4 PUBLIC "-//mybatis.org/DTD Mapper 3.0//EN"
6 5 "http://mybatis.org/dtd/mybatis-3-mapper.dtd">
7 6 <mapper namespace="member">
8 7 <select id="findMemberId" resultType="member">
9 8 SELECT id, password, name, address
10 9 FROM member
11 10 WHERE id=#{value}
12 11 </select>
13 12
14 13 <insert id="enrollMember" parameterType="com.springmybatis.dto.Member">
15 14 INSERT INTO Member VALUES(#{id}, #{password}, #{name}, #{address})
16 15 </insert>
17 16 </mapper>

```

```

1 MemberDaoImpl.java Test.java
2 package com.springmybatis.dao.impl;
3 import org.mybatis.spring.SqlSessionTemplate;
4
5 public class MemberDaoImpl implements MemberDao {
6
7     private SqlSessionTemplate sqlSessionTemplate;
8
9     public MemberDaoImpl(SqlSessionTemplate sqlSessionTemplate) {
10         super();
11         this.sqlSessionTemplate = sqlSessionTemplate;
12     }
13
14     @Override
15     public Member findMemberById(String id) {
16         return sqlSessionTemplate.selectOne(
17             "member.findMemberId", id);
18     }
19
20     @Override
21     public int insertMember(Member member) {
22         return sqlSessionTemplate.insert("member.enrollMember", member);
23     }
24 }

```

```

1 MemberDaoImpl.java member.xml Test.java
2 package com.springmybatis.test;
3 import org.springframework.context.ApplicationContext;
4
5 public class Test {
6
7     private static ApplicationContext context;
8
9     public static void main(String[] args) {
10         context = new ClassPathXmlApplicationContext("config.xml");
11         MemberDao memberDao =
12             (MemberDao) context.getBean("memberDao");
13         Member member = memberDao.findMemberById("python");
14         System.out.println(member);
15
16         member = new Member("cplusplus", "goodlang", " ", " ");
17         int result = memberDao.insertMember(member);
18         System.out.println("Number Records of input: " + result);
19
20     }
21 }

```

```

INFO: Loading XML bean definitions from class path resource [config.xml]
Member [id=python, password=python1234, name= , address= ]
Number Records of input: 1

```