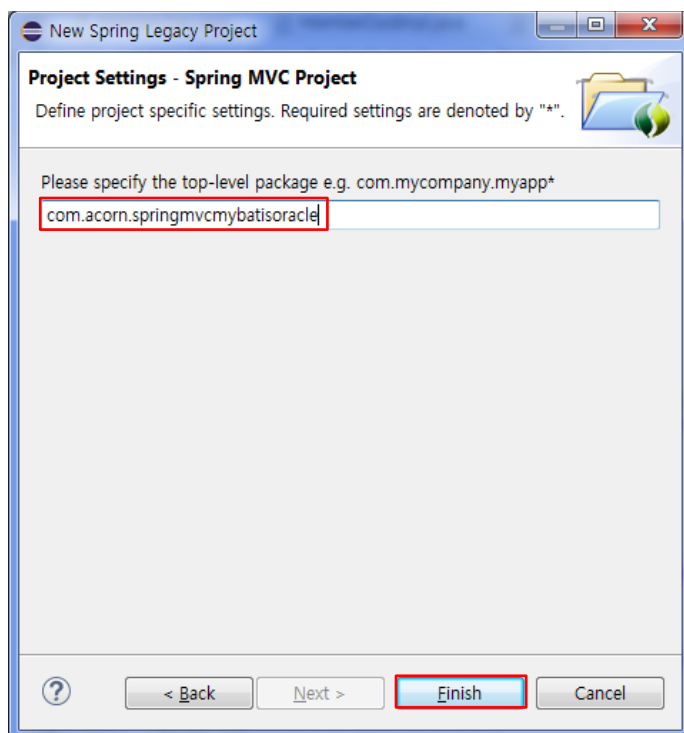
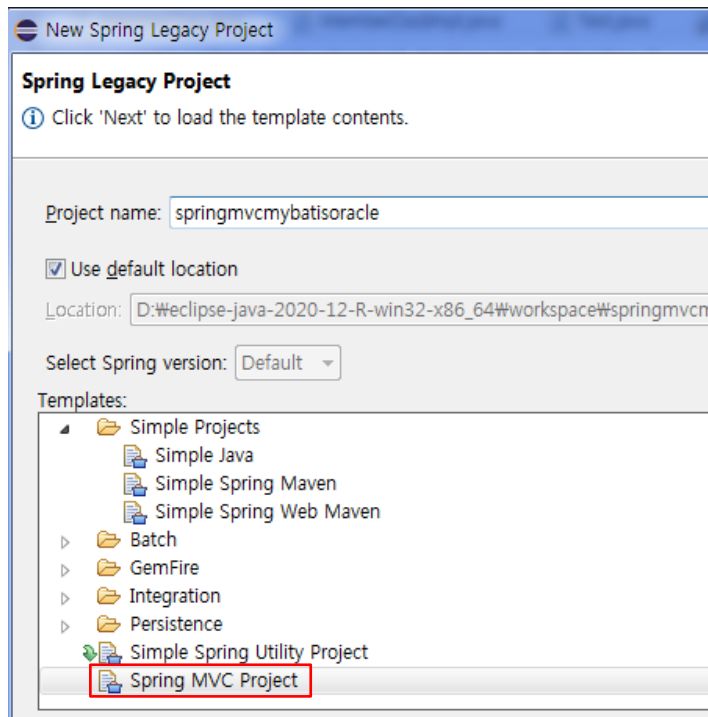
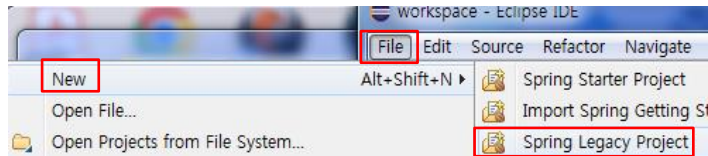


목차

- 0. Lombok/JPA 연관관계: Lombok 설치.pdf/JPA 연관관계실습.pdf/JPA 연관관계실습.mdj
- 1. Spring Legacy + JDBC/SPRINGJDBC + JSP
- 2. **Spring Legacy + MyBatis + JSP**
- 3. Spring Legacy + JPA + JSP
- 4. Spring boot + JDBC/SPRINGJDBC + JSP
- 5. Spring boot + MyBatis + JSP
- 6. Spring boot + JPA + Thymeleaf(Spring boot default)
- 7. Spring boot + MyBatis + Thymeleaf(Spring boot default)
- 8. **Spring boot + JPA + JSP**

2. Spring Legacy + MyBatis + JSP

2.1. 스프링 MVC 프로젝트 만들기



2.2. pom.xml 수정

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
```

```
  <modelVersion>4.0.0</modelVersion>
```

```
  <groupId>com</groupId>
```

```
  <artifactId>springmvcmybatisoracle</artifactId>
```

```
  <version>0.0.1-SNAPSHOT</version>
```

```
  <packaging>war</packaging>
```

```
  <name>springmvcmybatisoracle</name>
```

```
  <!-- FIXME change it to the project's website -->
```

```
  <url>http://www.example.com</url>
```

```
  <properties>
```

```
    <springframework.version>4.3.6.RELEASE</springframework.version>
```

```
  </properties>
```

```
  <dependencies>
```

```
    <dependency>
```

```
      <groupId>org.springframework</groupId>
```

```
      <artifactId>spring-webmvc</artifactId>
```

```
      <version>${springframework.version}</version>
```

```
    </dependency>
```

```
    <dependency>
```

```
      <groupId>org.springframework</groupId>
```

```
      <artifactId>spring-jdbc</artifactId>
```

```
      <version>${springframework.version}</version>
```

```
    </dependency>
```

```
    <dependency>
```

```
      <groupId>org.mybatis</groupId>
```

```
      <artifactId>mybatis</artifactId>
```

```
      <version>3.5.4</version>
```

```
    </dependency>
```

```
    <dependency>
```

```
      <groupId>org.mybatis</groupId>
```

```
      <artifactId>mybatis-spring</artifactId>
```

```
      <version>2.0.5</version>
```

```
    </dependency>
```

```

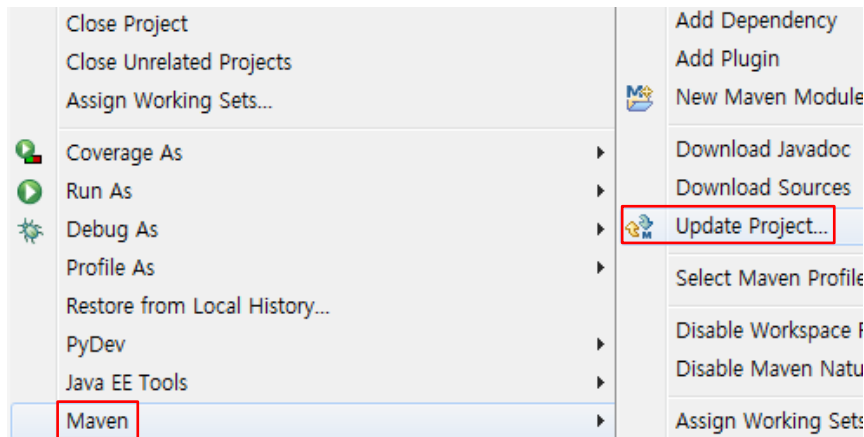
    <dependency>
      <groupId>org.apache.commons</groupId>
      <artifactId>commons-dbcp2</artifactId>
      <version>2.7.0</version>
    </dependency>

    <dependency>
      <groupId>com.oracle.database.jdbc</groupId>
      <artifactId>ojdbc6</artifactId>
      <version>11.2.0.4</version>
    </dependency>

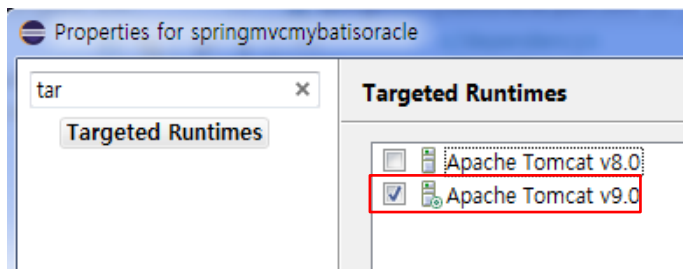
    <dependency>
      <groupId>jstl</groupId>
      <artifactId>jstl</artifactId>
      <version>1.2</version>
    </dependency>
  </dependencies>
  <build>
    <pluginManagement>
      <plugins>
        <plugin>
          <groupId>org.apache.maven.plugins</groupId>
          <artifactId>maven-compiler-plugin</artifactId>
          <version>3.8.0</version>
          <configuration>
            <source>11</source>
            <target>11</target>
          </configuration>
        </plugin>
      </plugins>
    </pluginManagement>
  </build>
</project>

```

2.3. maven project update



2.4. 아파치 런타임 추가



2.5. 소스코드에서 로그 패키지 및 코드 제거

```

1 package com.acorn.springmvcmybatisoracle;
2
3 import java.text.DateFormat;
4 import java.util.Date;
5 import java.util.Locale;
6
7 import org.slf4j.Logger;
8 import org.slf4j.LoggerFactory;
9 import org.springframework.stereotype.Controller;
10 import org.springframework.ui.Model;
11 import org.springframework.web.bind.annotation.RequestMapping;
12 import org.springframework.web.bind.annotation.RequestMethod;
13
14 /**
15  * Handles requests for the application home page.
16  */
17 @Controller
18 public class HomeController {
19
20     private static final Logger logger = LoggerFactory.getLogger(HomeController.class);
21
22     /**
23      * Simply selects the home view to render by returning its name.
24      */
25     @RequestMapping(value = "/", method = RequestMethod.GET)
26     public String home(Locale locale, Model model) {
27         logger.info("Welcome home! The client locale is {}.", locale);
28     }
29 }

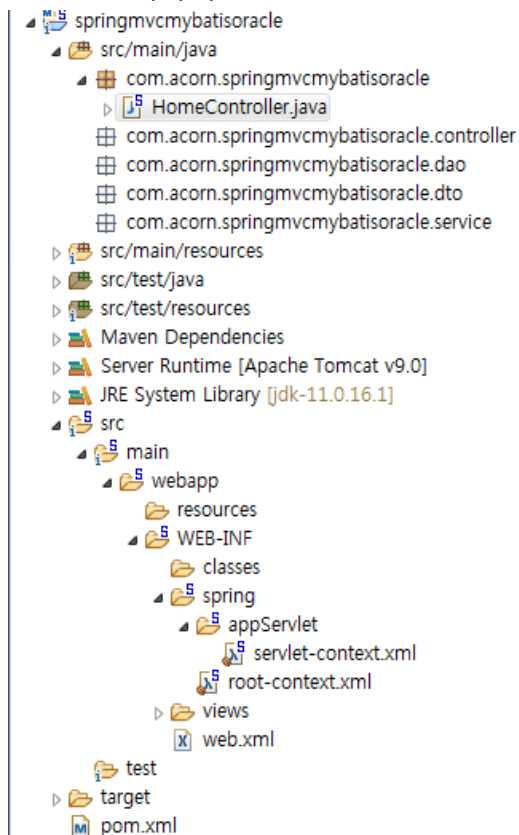
```

```

1 package com.acorn.springmvcmybatisoracle;
2
3 import java.text.DateFormat;
4 import java.util.Date;
5 import java.util.Locale;
6
7 import org.springframework.stereotype.Controller;
8 import org.springframework.ui.Model;
9 import org.springframework.web.bind.annotation.RequestMapping;
10 import org.springframework.web.bind.annotation.RequestMethod;
11
12 /**
13  * Handles requests for the application home page.
14  */
15 @Controller
16 public class HomeController {
17
18     @RequestMapping(value = "/", method = RequestMethod.GET)
19     public String home(Locale locale, Model model) {
20
21         Date date = new Date();
22         DateFormat dateFormat = DateFormat.getDateInstance(D
23
24         String formattedDate = dateFormat.format(date);
25
26         model.addAttribute("serverTime", formattedDate );
27
28         return "home";
29     }
30
31 }

```

2.6. 현재까지 완료된 모습



2.7. Spring Legacy MVC 프로젝트가 자동으로 생성한 웹 관련 설정 확인

2.8. MyBatis 설정: root-context.xml 파일에 데이터소스와 MyBatis 연동 설정

```
root-context.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="oracle.jdbc.OracleDriver"
16         p:url="jdbc:oracle:thin:@localhost:1521:xe"
17         p:username="oracle_test" p:password="woseven" />
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.acorn.springmvcmybatisoracle.dto" />
25
26     <bean
27         class="org.mybatis.spring.SqlSessionTemplate"
28         name="sqlSessionTemplate"
29         c:sqlSessionFactory-ref="sqlSessionFactoryBean" />
30
31 </beans>
```

2.9. SQL MAPPER 개발: classpath(src/main/java)/mybatis/config/member.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Sql Mapper -->
<!DOCTYPE mapper
PUBLIC "-//mybatis.org/DTD Mapper 3.0//EN"
"http://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="member">

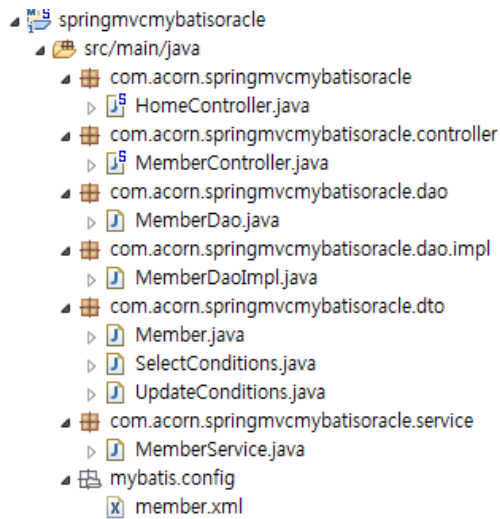
    <select id="findMemberById" parameterType="int" resultType="member">
    <![CDATA[
        SELECT    id, password, name, address
        FROM      member_new
        WHERE     id=#{id}
    ]]>
    </select>
</mapper>

<bean
class="org.apache.commons.dbcp2.BasicDataSource"
name="dbcp"
p:driverClassName="oracle.jdbc.OracleDriver"
p:url="jdbc:oracle:thin:@localhost:1521:xe"
p:username="oracle_test" p:password="woseven" />

<bean
class="org.mybatis.spring.SqlSessionFactoryBean"
name="sqlSessionFactoryBean"
p:dataSource-ref="dbcp"
p:mapperLocations="classpath:/mybatis/config/*.xml"
p:typeAliasesPackage="com.acorn.springmvcmybatisoracle.dto" />
```

```
springmvcmybatisoracle
├── src/main/java
│   └── com.acorn.springmvcmybatisoracle
│       ├── HomeController.java
│       ├── com.acorn.springmvcmybatisoracle.controller
│       ├── com.acorn.springmvcmybatisoracle.dao
│       ├── com.acorn.springmvcmybatisoracle.dto
│       └── com.acorn.springmvcmybatisoracle.service
└── mybatis.config
    └── member.xml
```

2.10. 클래스 작성: Controller/Service/Dto/Dao



가. MemberController.java

```
package com.acorn.springmvcmybatisoracle.controller;
```

```
@Controller
```

```
public class MemberController {
```

```
    @Autowired
```

```
    MemberService memberService;
```

```
    @RequestMapping (value = "/findMemberById", method = RequestMethod.GET)
```

```
    public String findMemberById(
```

```
        @RequestParam("id") String id, Model model) {
```

```
        Member member =
```

```
            memberService.findMemberById(Integer.valueOf(id));
```

```
        model.addAttribute("member", member);
```

```
        return "memberOne";
```

```
    }
```

```
}
```

나. MemberService.java

```
package com.acorn.springmvcmybatisoracle.service;
```

```
@Service
```

```
public class MemberService {
```

```
    @Autowired
```



```

        MemberDao memberDao;

        public Member findMemberById(int id) {

            Member member = memberDao.findMemberById(id);
            return member;
        }
    }
}

```

㉔. MemberDao.java

package com.acorn.springmvcmybatisoracle.dao;

```

public interface MemberDao {
    public Member findMemberById(int id);
}

```

㉕. MemberDaoImpl.java

package com.acorn.springmvcmybatisoracle.dao.impl;

```

@Repository
public class MemberDaoImpl implements MemberDao {
    @Autowired
    private SqlSessionTemplate sqlSessionTemplate;

    @Override
    public Member findMemberById(int id) {
        return sqlSessionTemplate.selectOne(
            "member.findMemberById", id);
    }
}

```

㉖. Member.java

package com.acorn.springmvcmybatisoracle.dto;

```

@Getter
@Setter
@AllArgsConstructor
@NoArgsConstructor
@ToString
public class Member {
    private int id;
    private String password;
    private String name;
    private String address;
}

```

㉗. memberOne.jsp

```







































<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8" isELIgnored="false" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>JSP JSPList</title>
  </head>
  <body>
    <table border="1">
      <tr>
        <th>아이디 </th>
        <th>패스워드 </th>
        <th>이름 </th>
        <th>주소 </th>
      </tr>
      <tr>
        <td>${member.id}</td>
        <td>${member.password}</td>
        <td>${member.name}</td>
        <td>${member.address}</td>
      </tr>
    </table>
  </body>
</html>
```

2.11. 추가 작업: SELECTONE/SELECTLIST/INSERT/UPDATE

2.12. 완성된 프로젝트

- ▲  springmvcmybatisoracle
 - ▲  src/main/java
 - ▲  com.acorn.springmvcmybatisoracle.controller
 - ▷  MemberController.java
 - ▲  com.acorn.springmvcmybatisoracle.dao
 - ▷  MemberDao.java
 - ▲  com.acorn.springmvcmybatisoracle.dao.impl
 - ▷  MemberDaoImpl.java
 - ▲  com.acorn.springmvcmybatisoracle.dto
 - ▷  Member.java
 - ▷  SelectConditions.java
 - ▲  com.acorn.springmvcmybatisoracle.service
 - ▷  MemberService.java
 - ▲  mybatis.config
 - ▷  member.xml
 - ▷  src/main/resources
 - ▷  src/test/java
 - ▷  src/test/resources
 - ▷  Maven Dependencies
 - ▷  Server Runtime [Apache Tomcat v9.0]
 - ▷  JRE System Library [JavaSE-11]
 - ▲  src
 - ▲  main
 - ▲  webapp
 -  resources
 - ▲  WEB-INF
 -  classes
 - ▲  spring
 - ▲  appServlet
 - ▷  servlet-context.xml
 - ▷  root-context.xml
 - ▲  views
 - ▷  memberList.jsp
 - ▷  memberOne.jsp
 - ▷  web.xml
 - ▷  test
 - ▷  target
 - ▷  pom.xml

2.13. 예외처리: INSERT(기본키)/Delete(외래키)/Update(외래키)

```
package com.acorn.springmvcmybatisoracle.exception;
```

```
import java.sql.SQLException;
```

```
import org.springframework.dao.DataAccessException;
```

```
import org.springframework.web.bind.annotation.ControllerAdvice;
```

```
import org.springframework.web.bind.annotation.ExceptionHandler;
```

```
import org.springframework.web.servlet.ModelAndView;
```

```
@ControllerAdvice
```

```
public class ExceptionController {
```

```
    @ExceptionHandler({SQLException.class, DataAccessException.class})
```

```
    public ModelAndView handleSQLException(Exception e) {
```

```
        ModelAndView model = new ModelAndView();
```

```
        model.addObject("code", "500");
```

```
        model.addObject("msg", "데이터베이스 입력 작업 중 오류 발생");
```

```
        model.setViewName("insertMember");
```

```
        return model;
```

```
    }
```

```
}
```

