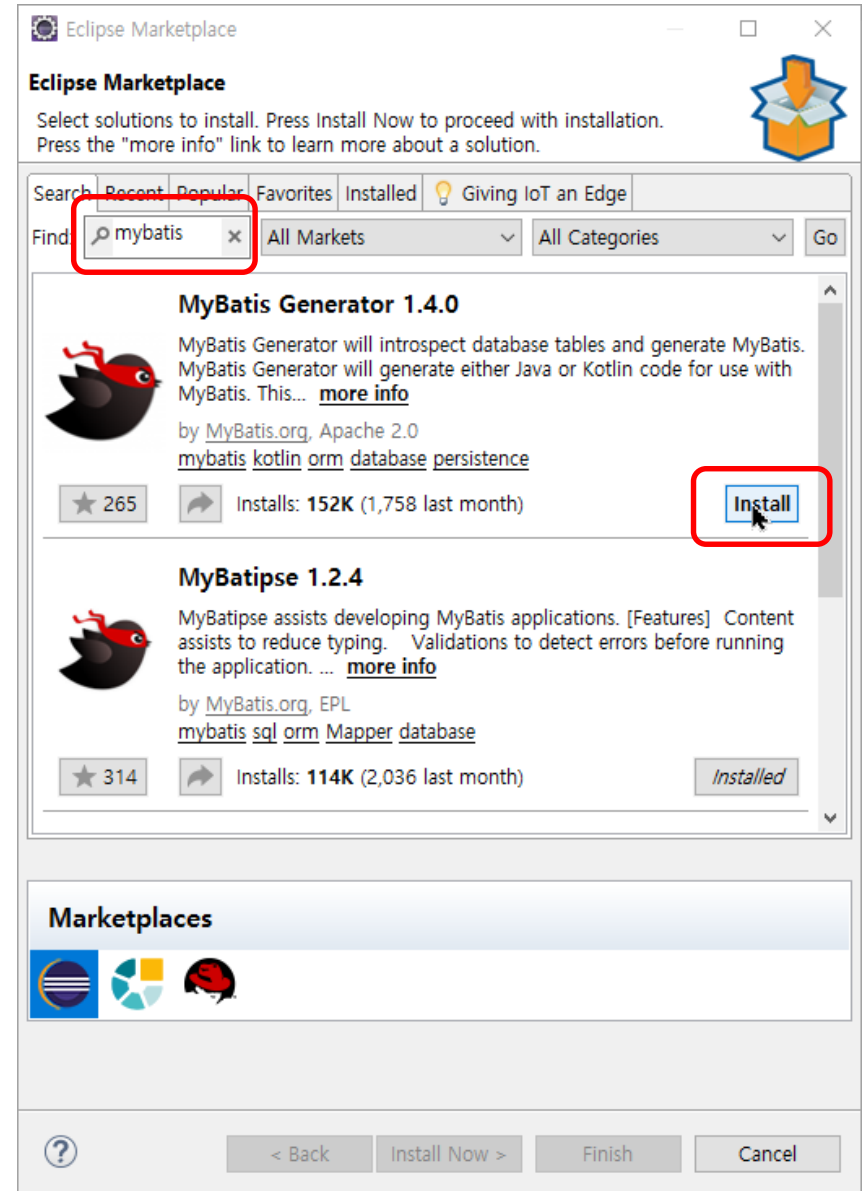
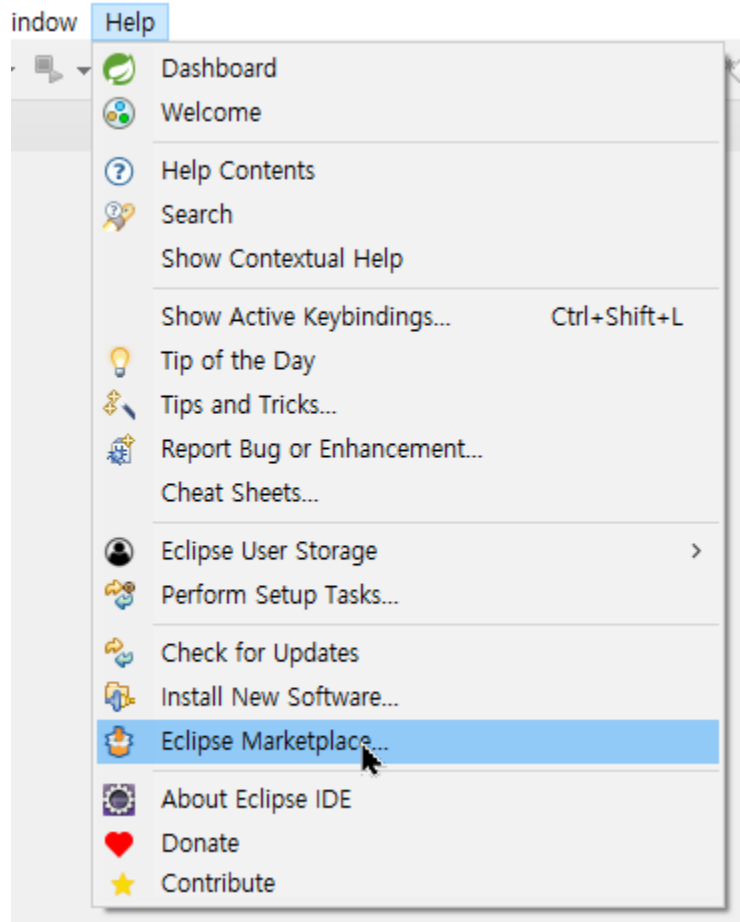


MyBatis

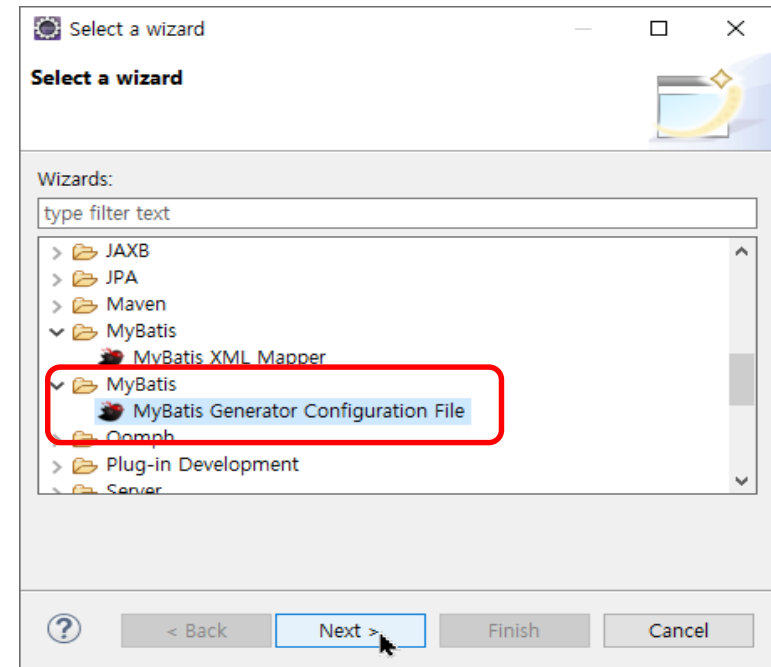
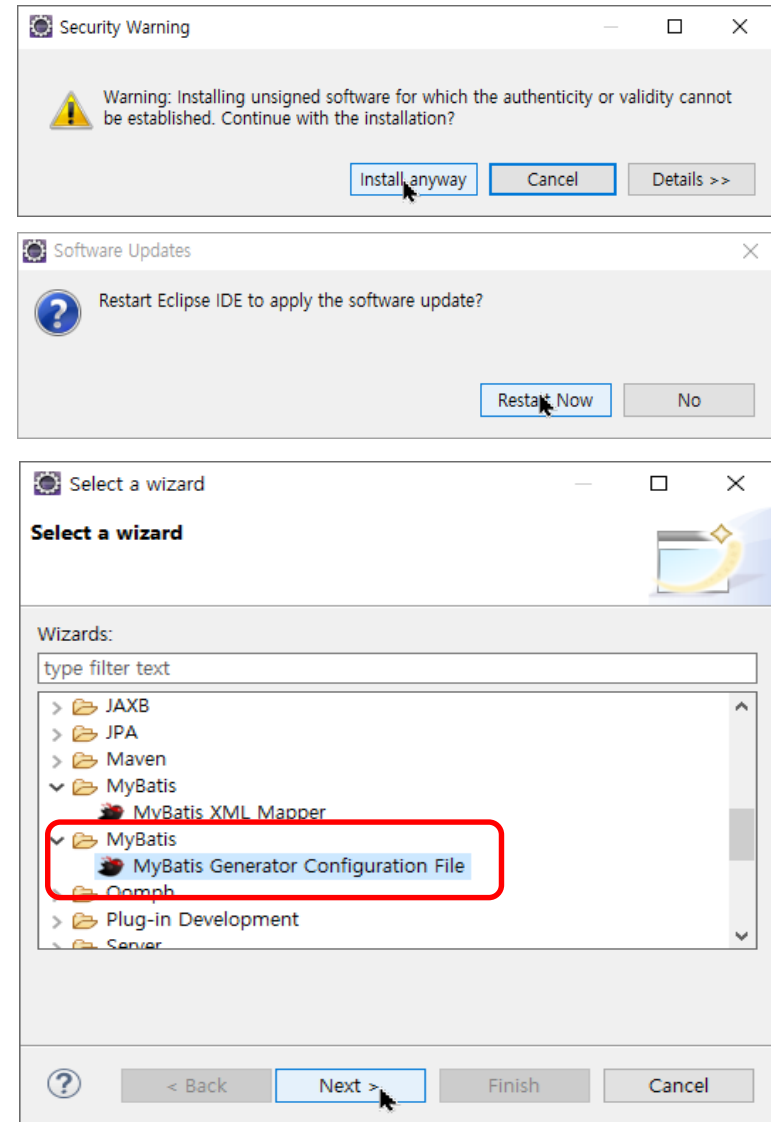
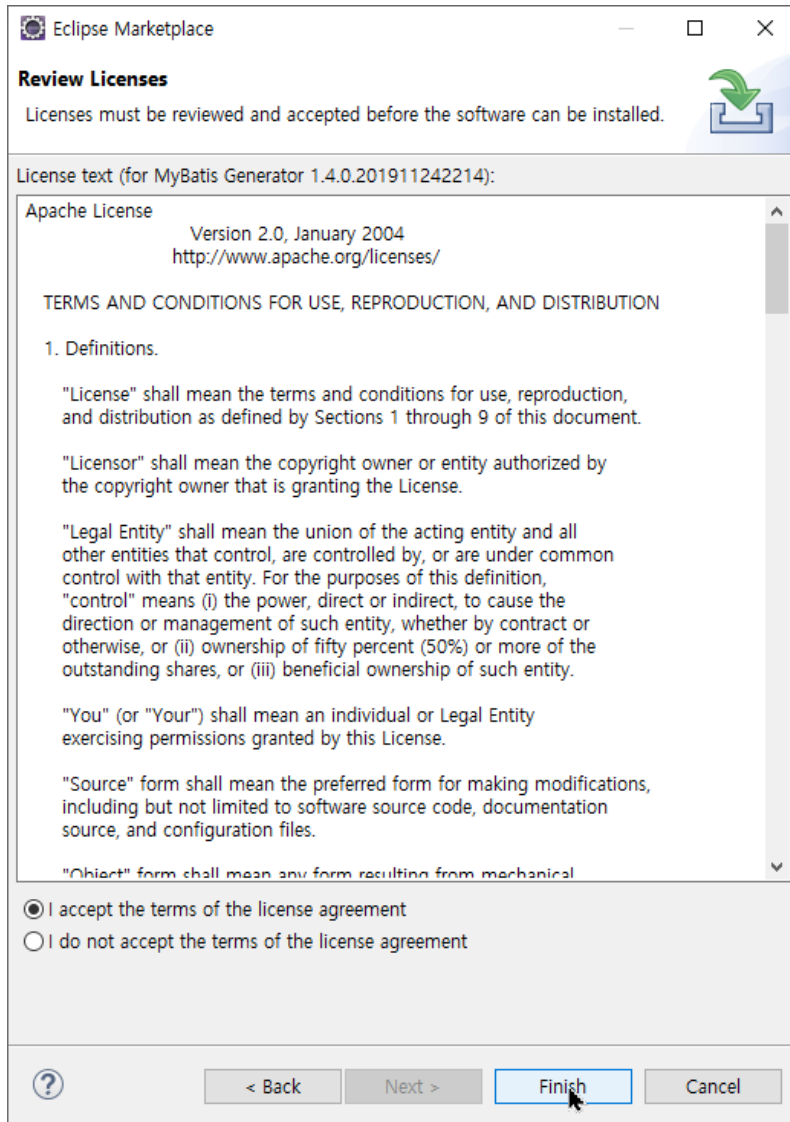
MyBatis 프레임워크

- 홈페이지 : <https://mybatis.org/mybatis-3/>
- Apache에서 iBatis라는 이름의 프레임워크로 탄생.
- 2010년 iBatis가 Apache에서 탈퇴하여 Google로 넘어가면서 이름이 MyBatis로 변경.
- 한두 줄의 자바 코드로 DB 연동을 처리.
- SQL 명령어를 자바 코드에서 분리하여 XML 파일에 따로 관리.
- XML 파일에 저장된 SQL 명령어를 대신 실행하고, 실행 결과를 VO 같은 자바 객체에 자동으로 매핑까지 해준다고 해서 데이터 매퍼(Data Mapper)라고 부름.

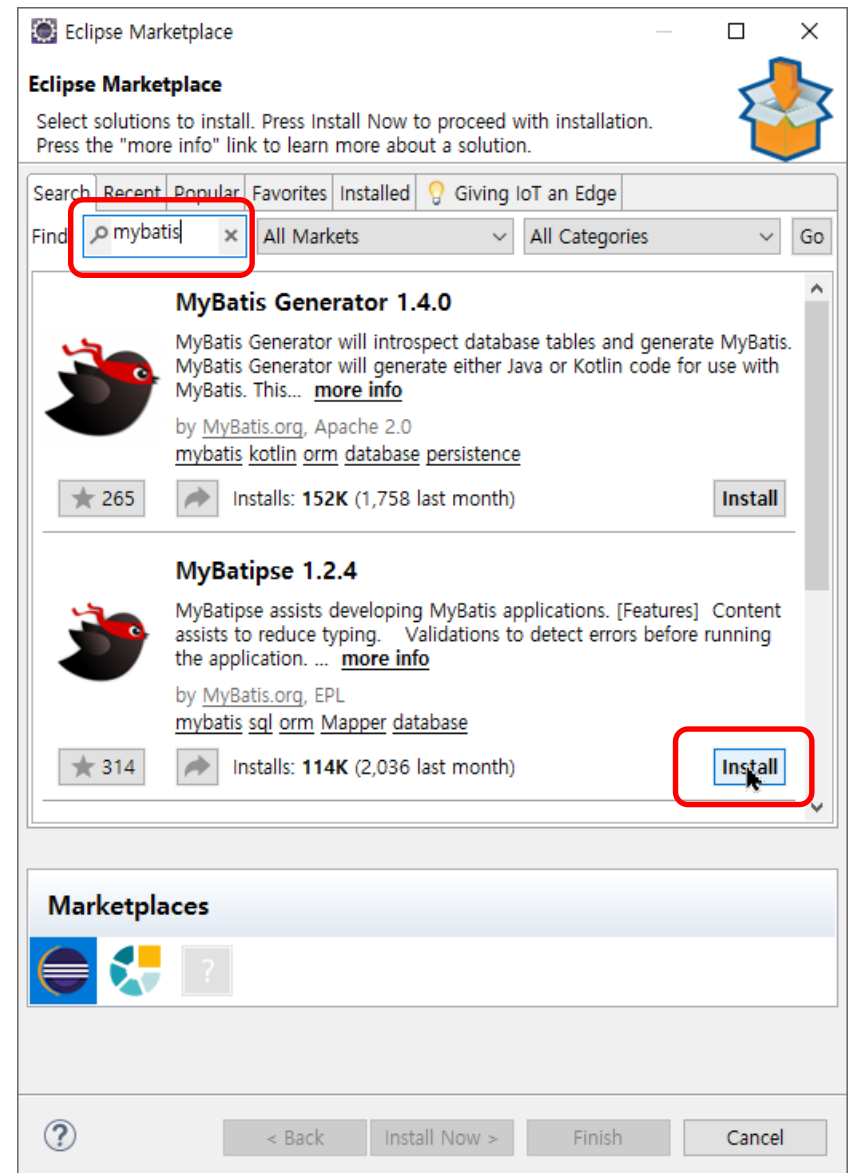
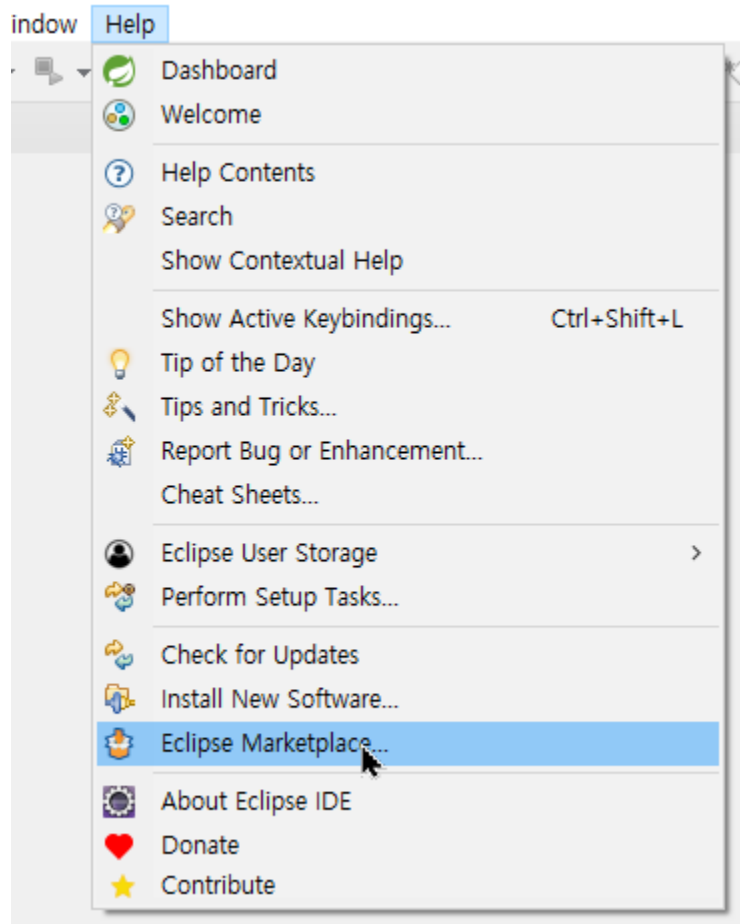
MyBatis Plugin 설치(1)



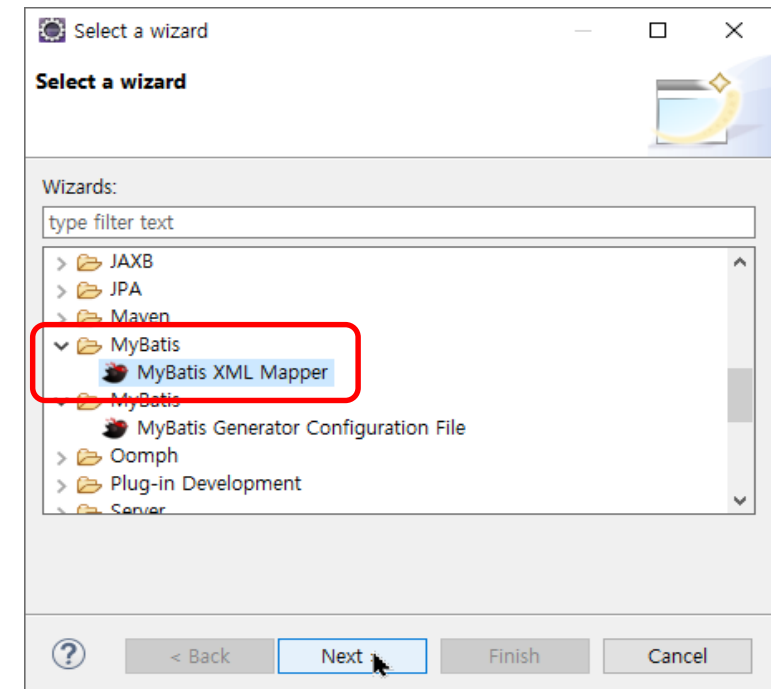
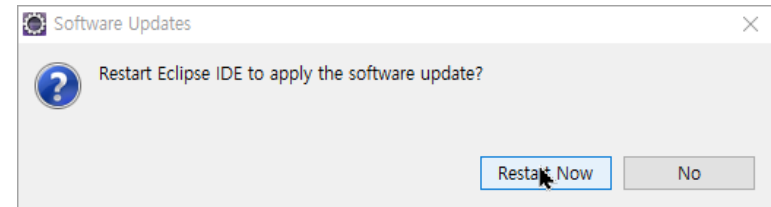
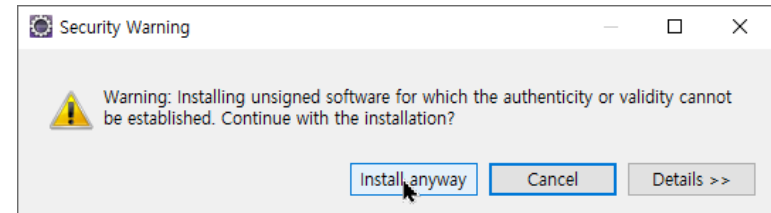
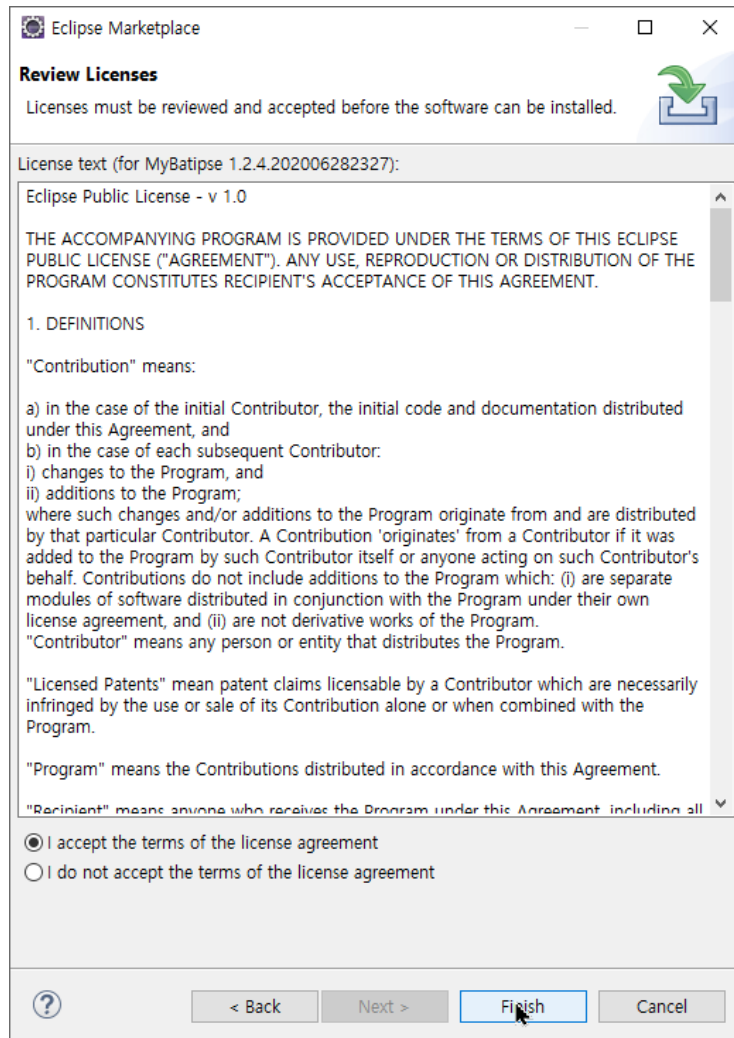
MyBatis Plugin 설치(1)



MyBatis Plugin 설치(2)








MyBatis Plugin 설치(2)



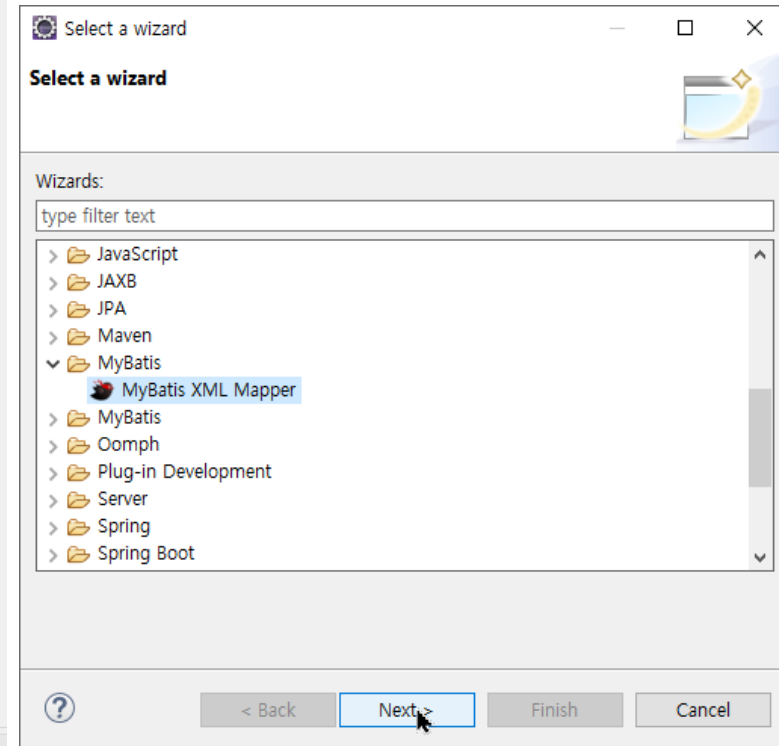
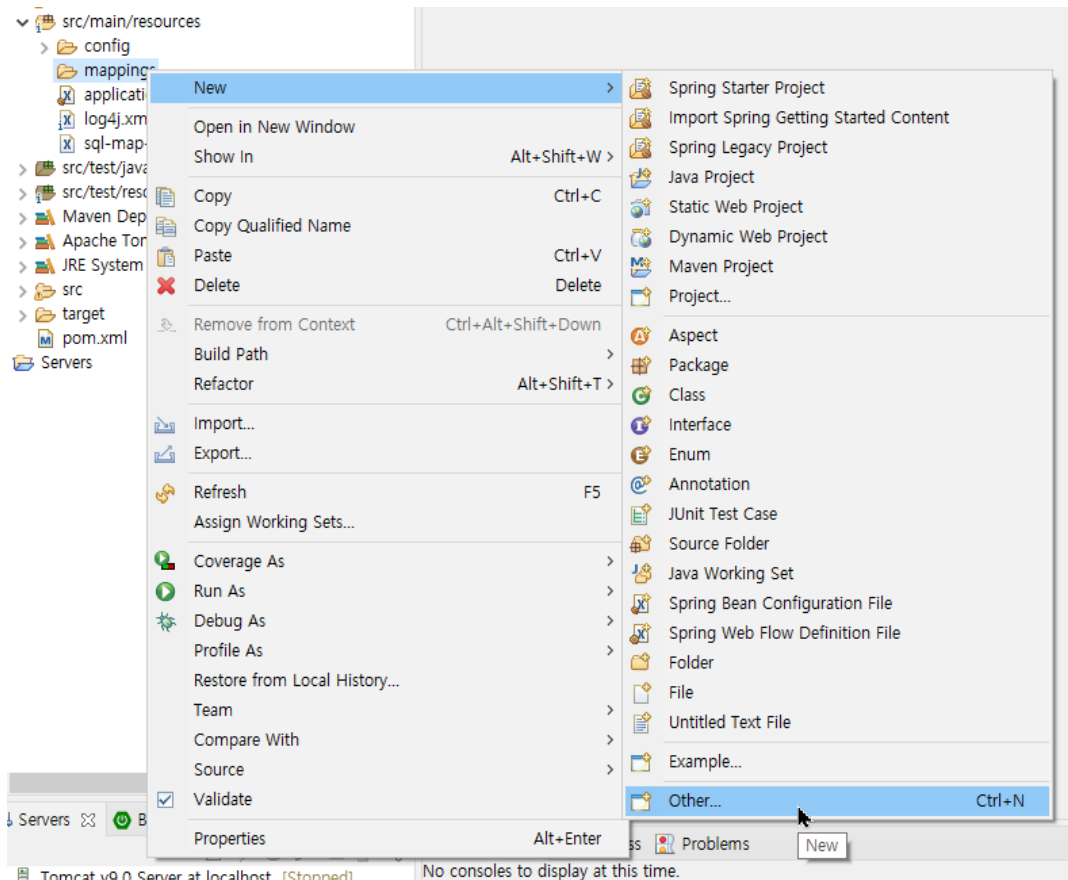
라이브러리 다운로드 받기(pom.xml 추가)

```
<!-- https://mvnrepository.com/artifact/org.mybatis/mybatis -->  
<dependency>  
  <groupId>org.mybatis</groupId>  
  <artifactId>mybatis</artifactId>  
  <version>3.5.6</version>  
</dependency>
```

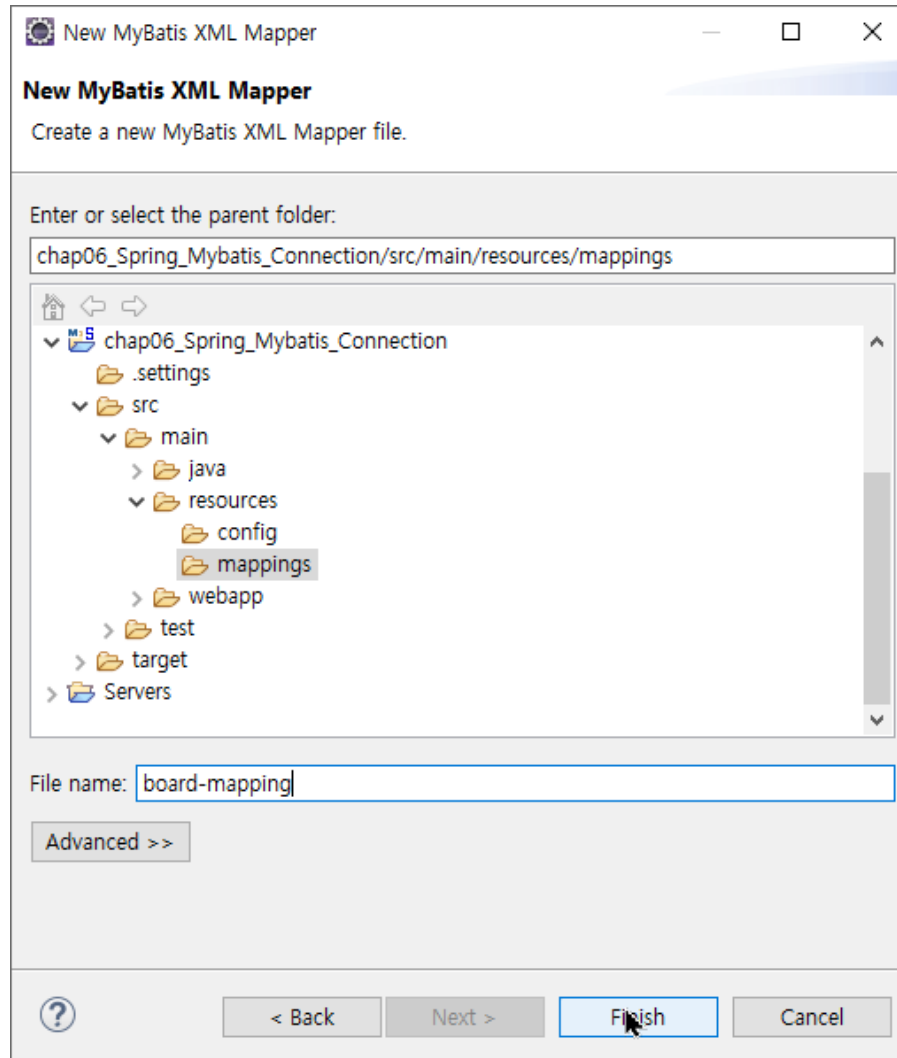
```
<!-- https://mvnrepository.com/artifact/org.mybatis/mybatis-spring -->  
<dependency>  
  <groupId>org.mybatis</groupId>  
  <artifactId>mybatis-spring</artifactId>  
  <version>2.0.6</version>  
</dependency>
```

- >  junit-4.7.jar - C:\Users\User\...
- >  log4j-1.2.15.jar - C:\Users\User\...
- >  mybatis-3.5.6.jar - C:\Users\User\...
- >  mybatis-spring-2.0.6.jar - C:\Users\User\...
- >  ojdbc6-11.2.0.4.jar - C:\Users\User\...

SQL Mapper XML 파일 작성



SQL Mapper XML 파일 작성

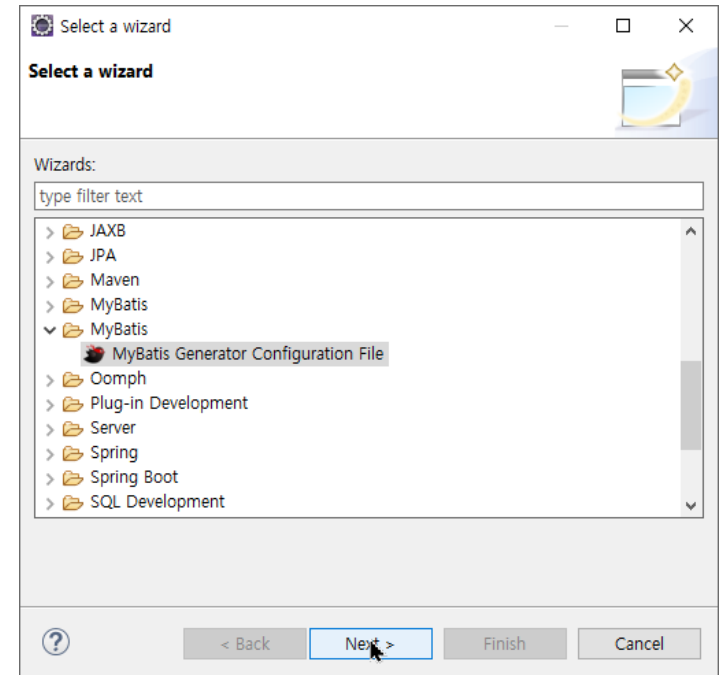
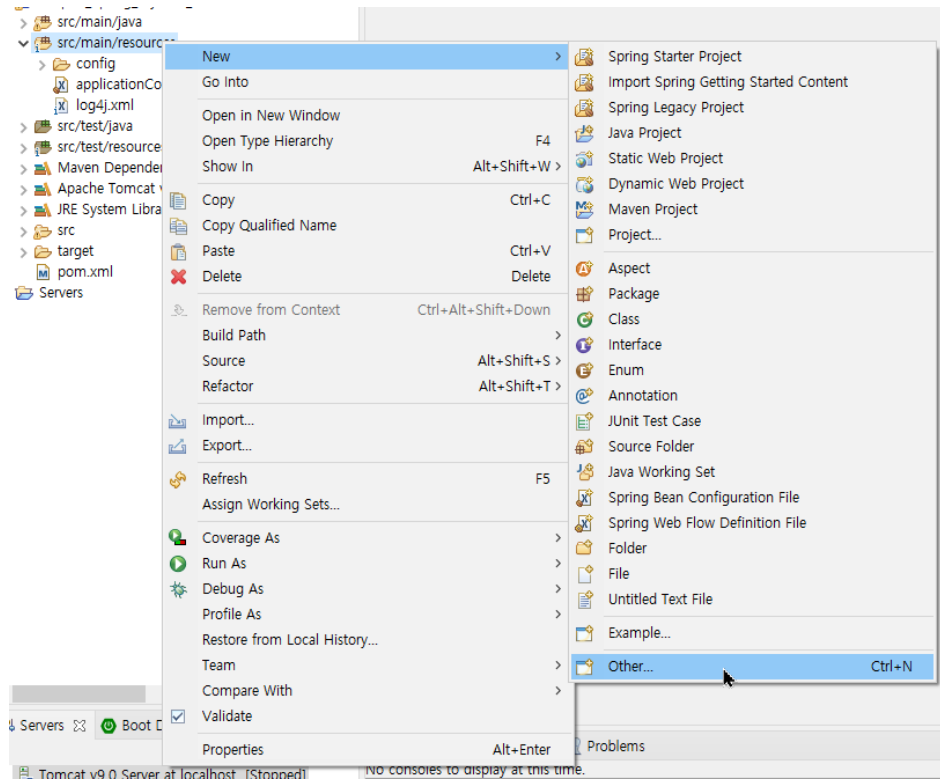


board-mapping.xml 작성

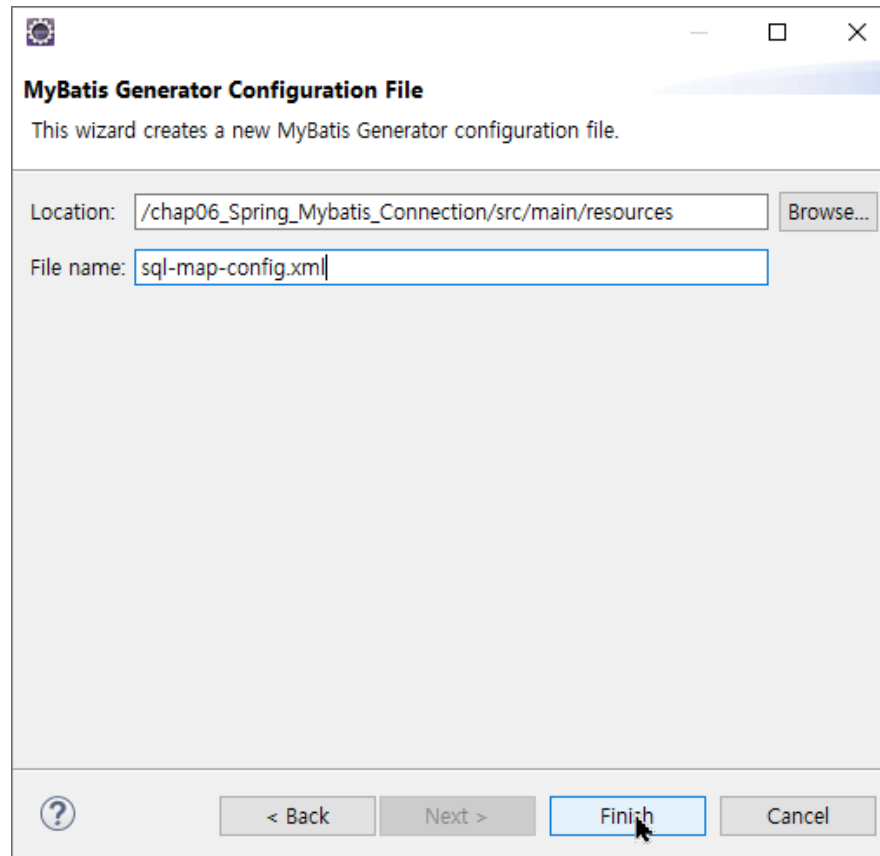
```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
    "http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="BoardDAO">
<insert id="insertBoard">
    insert into board(seq, title, writer, content)
    values((select nvl(max(seq), 0)+1 from board),#{title},#{writer},#{content})
</insert>
<update id="updateBoard">
    update board set title=#{title}, content=#{content} where seq=#{seq}
</update>
<delete id="deleteBoard">
    delete board where seq=#{seq}
</delete>
<select id="getBoard" resultType="board">
    select * from board where seq=#{seq}
</select>
<select id="getBoardList_T" resultType="board">
    select * from board
    where title like '%'||#{searchKeyword}||'%'
    order by seq desc
</select>
<select id="getBoardList_C" resultType="board">
    select * from board
    where content like '%'||#{searchKeyword}||'%'
    order by seq desc
</select>
</mapper>
```

MyBatis 환경 설정 파일 생성



MyBatis 환경 설정 파일 생성



sql-map-config.xml 수정

```
<?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>

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

    <!-- Sql Mapper 설정 -->
    <mappers>
        <mapper resource="mappings/board-mapping.xml" />
    </mappers>

</configuration>
```

스프링 연동 설정

- 스프링과 MyBatis를 연동하려면 우선 스프링 설정 파일에 SqlSessionFactoryBean 클래스를 bean 등록해야 함.
- SqlSessionFactoryBean 객체로부터 DB 연동 구현에 사용할 SqlSession 객체를 얻을 수 있음.

```
<!-- Spring과 Mybatis 연동 설정 -->  
<bean id="sqlSession" class="org.mybatis.spring.SqlSessionFactoryBean">  
    <property name="dataSource" ref="dataSource"/>  
    <property name="configLocation" value="classpath:sql-map-config.xml"/>  
</bean>
```

DAO 클래스 구현 - 방법1

- SqlSessionDaoSupport 클래스를 상속하여 구현하는 방법.

```
@Repository
public class BoardDAOMybatis extends SqlSessionDaoSupport {

    @Autowired
    public void setSqlSessionFactory(SqlSessionFactory sqlSessionFactory) {
        super.setSqlSessionFactory(sqlSessionFactory);
    }

    public void insertBoard(BoardVO vo) {
        System.out.println("==> Mybatis로 insertBoard() 기능 처리");
        getSqlSession().insert("BoardDAO.insertBoard", vo);
    }

    ....
}
```

DAO 클래스 구현 - 방법2

- SqlSessionTemplate 클래스를 <bean> 등록하여 사용하는 방법.
- SqlSessionTemplate 클래스에는 setter 메서드가 없어서 생성자 메서드를 이용한 constructor 주입으로 처리해야 함.
- DAO 클래스를 구현할 때는 SqlSessionTemplate 객체를 @Autowired하여 DB 연동 로직을 구현하면 됨.

```
<!-- applicationContext.xml 설정 파일 -->
```

```
<bean class="org.mybatis.spring.SqlSessionTemplate">  
    <constructor-arg ref="sqlSession"></constructor-arg>  
</bean>
```

```
@Repository  
public class BoardDAOMybatis {  
    @Autowired  
    private SqlSessionTemplate mybatis;  
  
    public void insertBoard(BoardVO vo) {  
        System.out.println("==> Mybatis로 insertBoard() 기능 처리");  
        mybatis.insert("BoardDAO.insertBoard", vo);  
    }  
  
    ....  
}
```