



mybatis

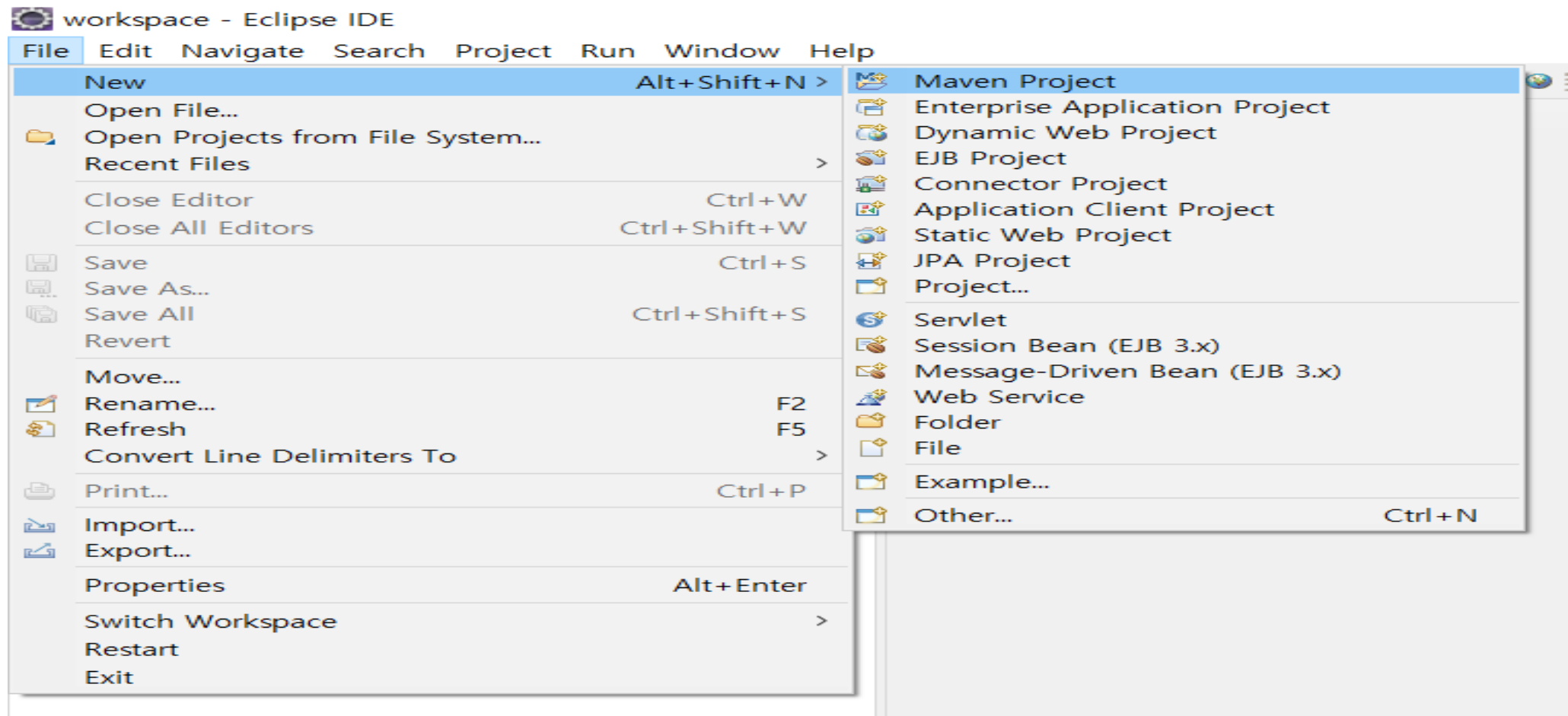
안화수

# 데이터베이스 접속 방식

1. JDBC(Java DataBase Connectivity) 방식
2. DBCP(DataBase Connection Pool) 방식
3. ORM(Object Relational Mapping) 프레임워크  
ex) iBatis, **MyBatis**, hibernate, JPA etc

# Maven Project 생성

## ❖ maven project



# Maven Project 생성

## ❖ maven project

New Maven Project

**New Maven project**

Select project name and location

☐ Create a simple project (skip archetype selection)

☒ Use default Workspace location

Location: C:\Users\giduc\git\myproject\jsppro Browse...

☐ Add project(s) to working set

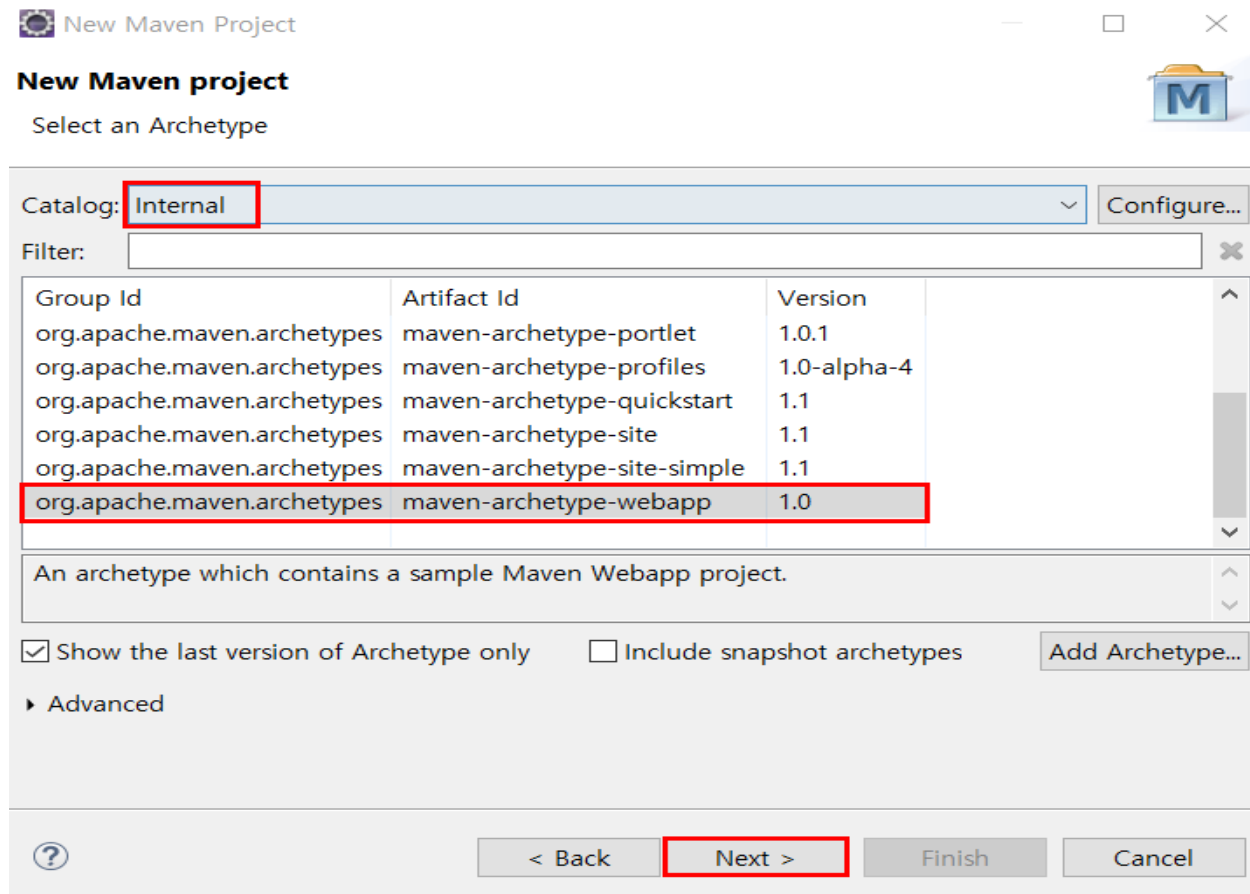
Working set: More...

▶ Advanced

? < Back **Next >** Finish Cancel

# Maven Project 생성

## ❖ maven project



# Maven Project 생성

## ❖ maven project

New Maven Project

**New Maven project**

Specify Archetype parameters

Group Id:

Artifact Id:

Version:

Package:

Properties available from archetype:

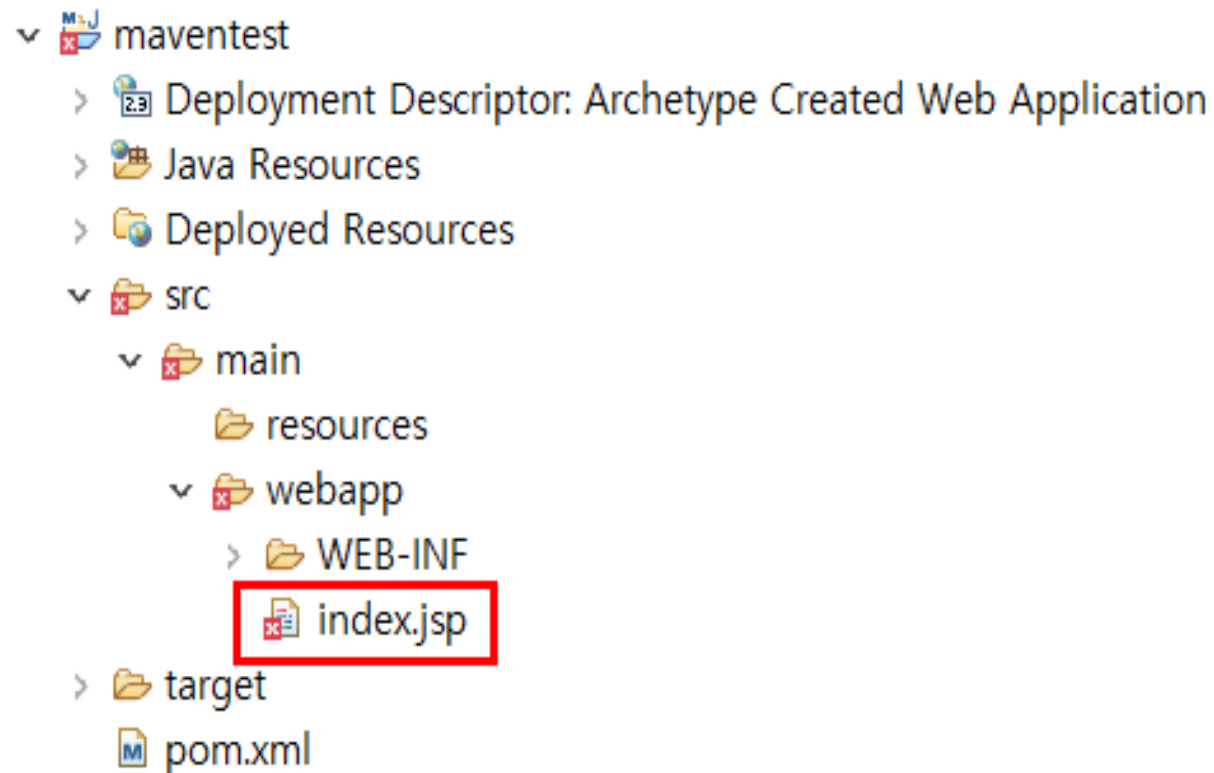
| Name | Value |
|------|-------|
|      |       |
|      |       |
|      |       |
|      |       |
|      |       |
|      |       |
|      |       |
|      |       |

Advanced

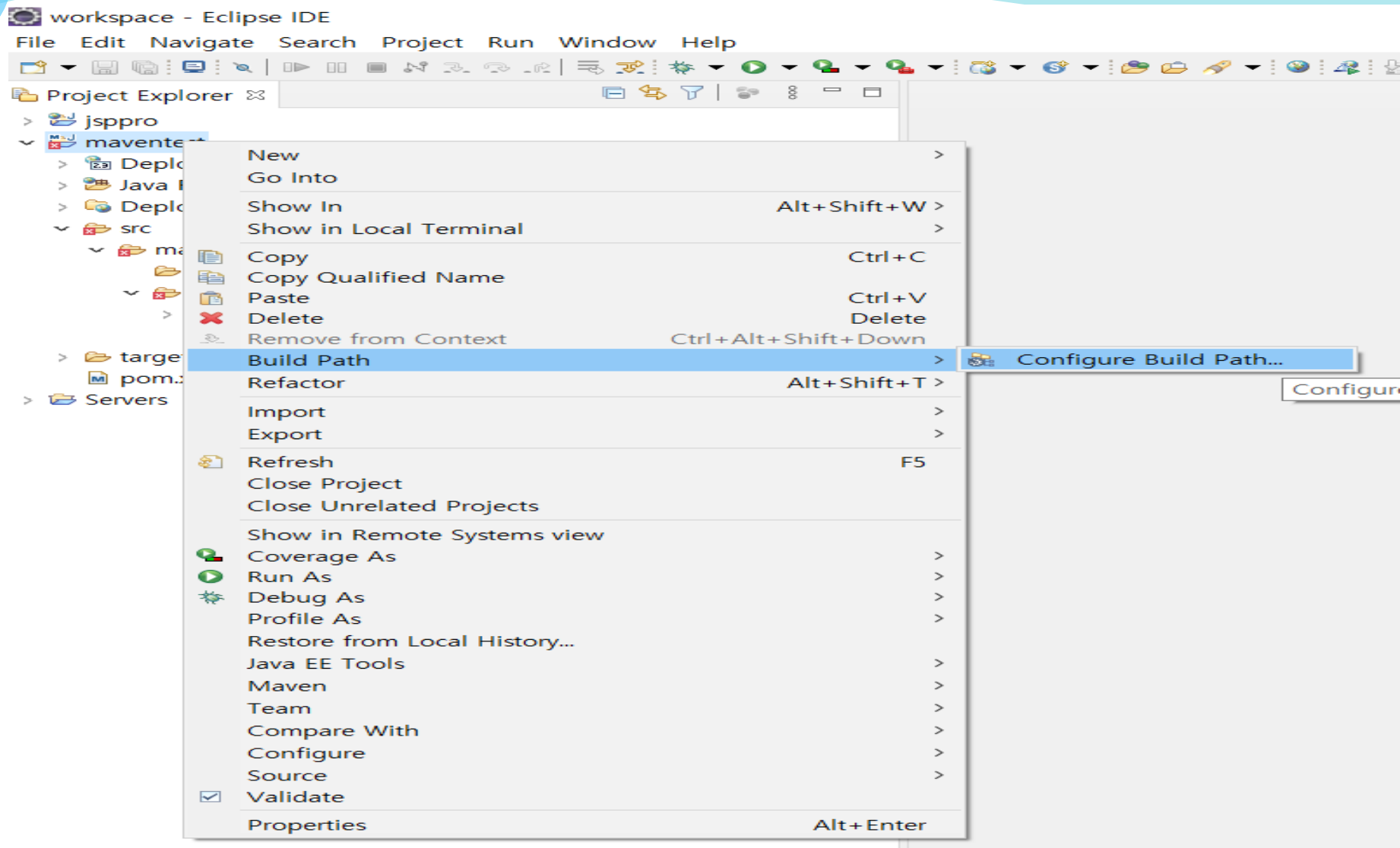
# Maven Project 생성

## ❖ maven project 에러발생

maven 프로젝트 생성후 index.jsp 파일에 에러가 발생하면, Apache Tomcat Library 를 추가하면 에러가 사라진다.

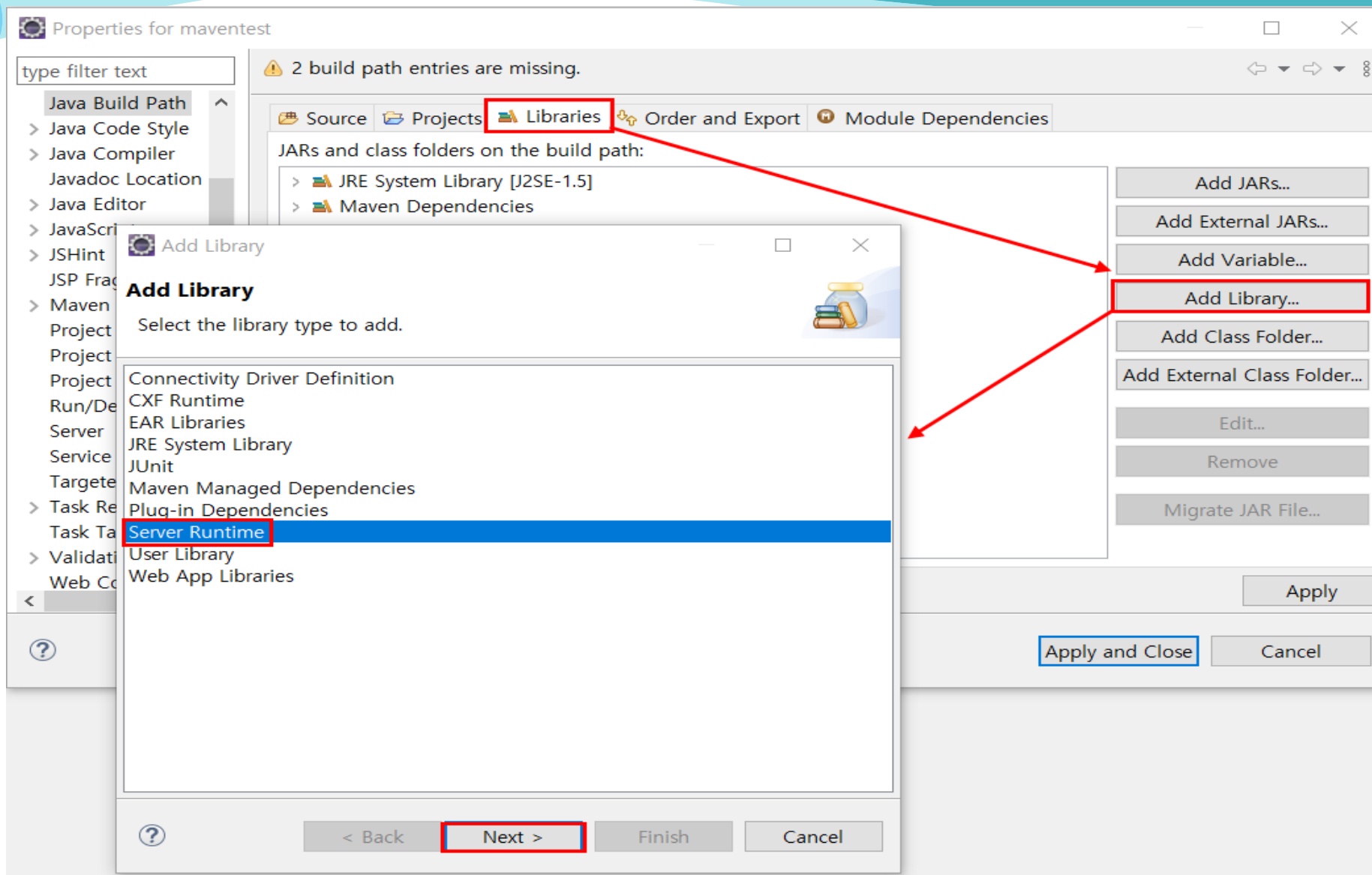


# Apache Tomcat Library 추가

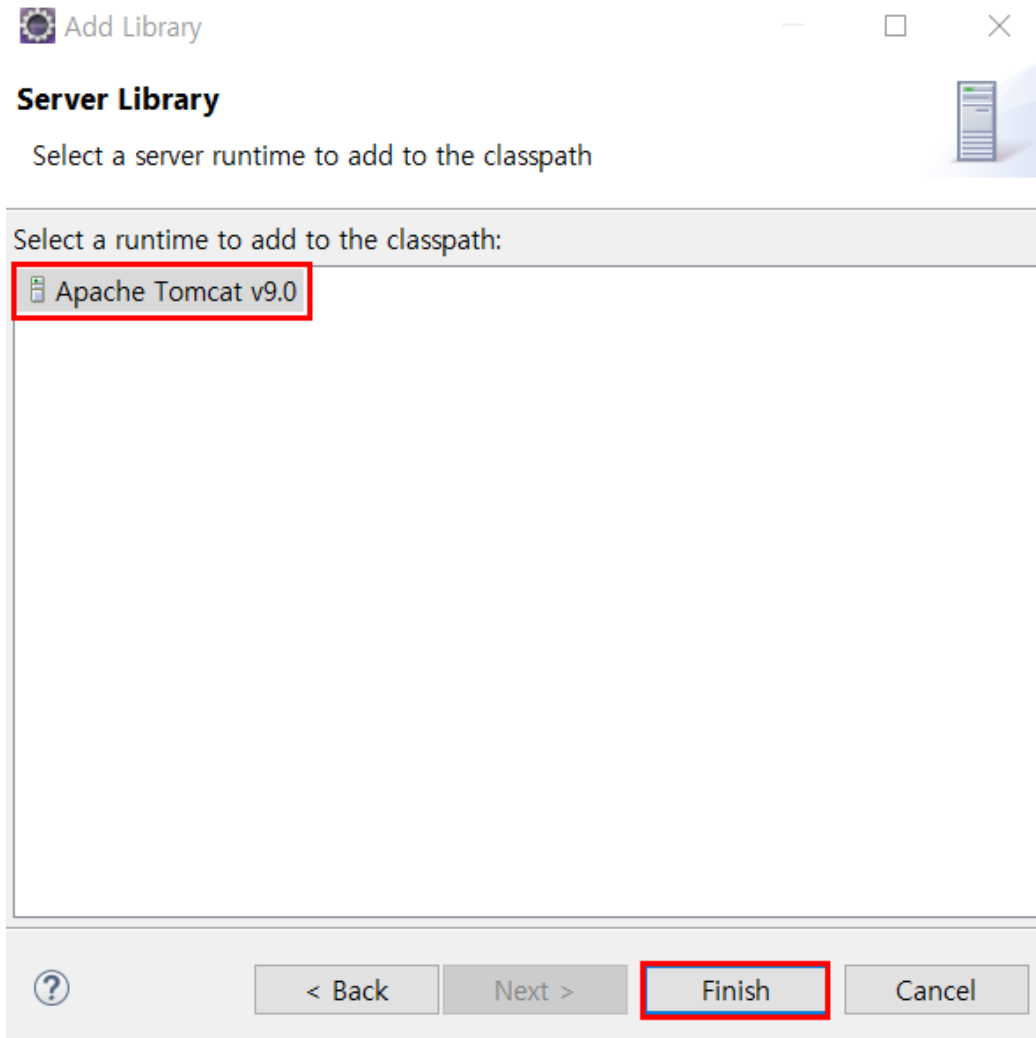




# Apache Tomcat Library 추가



# Apache Tomcat Library 추가



# Maven Project 구조

maventest - src - main - **java**

<--- Controller, Service, DTO, DAO

- **resources**

<--- MyBatis의 환경설정 파일, SQL문 파일(\*.xml)

- **webapp**

<--- view페이지

-WEB-INF - **web.xml** (project의 환경설정 파일)

- **pom.xml** (Maven 환경설정 파일)

# 의존 라이브러리 검색

❖ 의존 라이브러리 검색 ( cos 검색 )

<http://mvnrepository.com>

The screenshot shows the Maven Repository website with a search for 'cos'. The search bar at the top contains 'cos' and the 'Search' button is visible. The results page shows 'Found 188 results' and lists three items. The second item, 'COS' from 'servlets.com', is highlighted with a red box. The left sidebar shows a list of repositories, and the right sidebar shows a list of indexed repositories.

**Maven Repository: cos**

Search results for **cos**:

- Repository**
  - Central 176
  - Sonatype 8
  - JCenter 7
  - ImageJ Public 3
  - Spring Lib M 2
  - Spring Plugins 2
  - BeDataDriven 1
  - ICM 1
- Group**
  - de.dfki 61
  - data.de 58
  - com.ibm 17
  - com.qcloud 14
  - com.tencent 7
  - com.oldnoop 5
  - com.github 2
  - com.healthy-chn 2

**Found 188 results**

Sort: **relevance** | popular | newest

- 1. Cos**  
com.jfinal » cos  
Cos  
Last Release on Apr 11, 2020  
36 usages  
Apache
- 2. COS**  
servlets.com » cos  
The O'Reilly Servlet Package (com.oreilly.servlet) by Jason Hunter to accompany his book, Java Servlet Programming, including the famous MultipartRequest and MultipartParser classes.  
Last Release on Nov 8, 2005  
30 usages
- 3. COS**  
com.servlets » cos  
COS  
Last Release on Dec 20, 2005  
15 usages

**Indexed Repositories (1350)**

- Central
- Sonatype
- Atlassian
- Spring Plugins
- Spring Lib M
- Hortonworks
- JCenter
- Atlassian Public

# 의존 라이브러리 검색

## ❖ 의존 라이브러리 검색 ( cos 검색 )

[Home](#) » [servlets.com](#) » [cos](#)



### COS

The O'Reilly Servlet Package ([com.oreilly.servlet](#)) by Jason Hunter to accompany his book, Java Servlet Programming, including the famous `MultipartRequest` and `MultipartParser` classes.

|         |                              |
|---------|------------------------------|
| Tags    | <a href="#">servlet</a>      |
| Used By | <a href="#">30 artifacts</a> |

**Note:** This artifact was moved to:

[com.servlets](#) » [cos](#)

[Central \(1\)](#)

[Spring Plugins \(1\)](#)

| Version                   | Repository              | Usages             | Date      |
|---------------------------|-------------------------|--------------------|-----------|
| <a href="#">05Nov2002</a> | <a href="#">Central</a> | <a href="#">30</a> | Nov, 2005 |

# 의존 라이브러리 검색

## ❖ 의존 라이브러리 검색 ( cos 검색 )

[Home](#) » [servlets.com](#) » [cos](#) » 05Nov2002



### COS » 05Nov2002

The O'Reilly Servlet Package (com.oreilly.servlet) by Jason Hunter to accompany his book, Java Servlet Programming, including the famous MultipartRequest and MultipartParser classes.

|              |  |
|--------------|--|
| Organization | <a href="#">servlets.com</a>   |
| HomePage     | <a href="http://www.servlets.com/">http://www.servlets.com/</a>                      |
| Date         | (Nov 08, 2005)   |
| Files        | <a href="#">pom</a> (768 bytes) <a href="#">jar</a> (55 KB) <a href="#">View All</a> |
| Repositories | <a href="#">Central</a>  |
| Used By      | <b>30 artifacts</b>  |

[Maven](#) [Gradle](#) [Gradle \(Short\)](#) [Gradle \(Kotlin\)](#) [SBT](#) [Ivy](#) [Grape](#) [Leiningen](#)  
[Buildr](#)

```
<!-- https://mavenrepository.com/artifact/servlets.com/cos -->
<dependency>
  <groupId>servlets.com</groupId>
  <artifactId>cos</artifactId>
  <version>05Nov2002</version>
</dependency>
```

☒ Include comment with link to declaration

# Maven 환경설정 파일 : pom.xml

❖ pom.xml : cos 추가

```
<dependencies>
```

```
    <dependency>
```

```
        <groupId>servlets.com</groupId>
```

```
        <artifactId>cos</artifactId>
```

```
        <version>05Nov2002</version>
```

```
    </dependency>
```

```
</dependencies>
```

# Maven 환경설정 파일 : pom.xml

❖ pom.xml : oracle 추가

```
<dependencies>
```

```
    <dependency>
```

```
        <groupId>com.oracle.database.jdbc</groupId>
```

```
        <artifactId>ojdbc8</artifactId>
```

```
        <version>21.5.0.0</version>
```

```
    </dependency>
```

```
</dependencies>
```



# Maven 환경설정 파일 : pom.xml

❖ pom.xml : mysql 추가

```
<dependencies>
```

```
    <dependency>
```

```
        <groupId>com.mysql</groupId>
```

```
        <artifactId>mysql-connector-j</artifactId>
```

```
        <version>8.0.33</version>
```

```
    </dependency>
```

```
</dependencies>
```

# Maven 환경설정 파일 : pom.xml

❖ pom.xml : mybatis 추가

```
<dependencies>
```

```
    <dependency>
```

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

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

```
        <version>3.5.6</version>
```

```
    </dependency>
```

```
</dependencies>
```

# Maven 환경설정 파일 : pom.xml

❖ pom.xml : jstl 추가

```
<dependencies>
```

```
    <dependency>
```

```
        <groupId>javax.servlet</groupId>
```

```
        <artifactId>jstl</artifactId>
```

```
        <version>1.2</version>
```

```
    </dependency>
```

```
    <dependency>
```

```
        <groupId>org.glassfish.web</groupId>
```

```
        <artifactId>jstl-impl</artifactId>
```

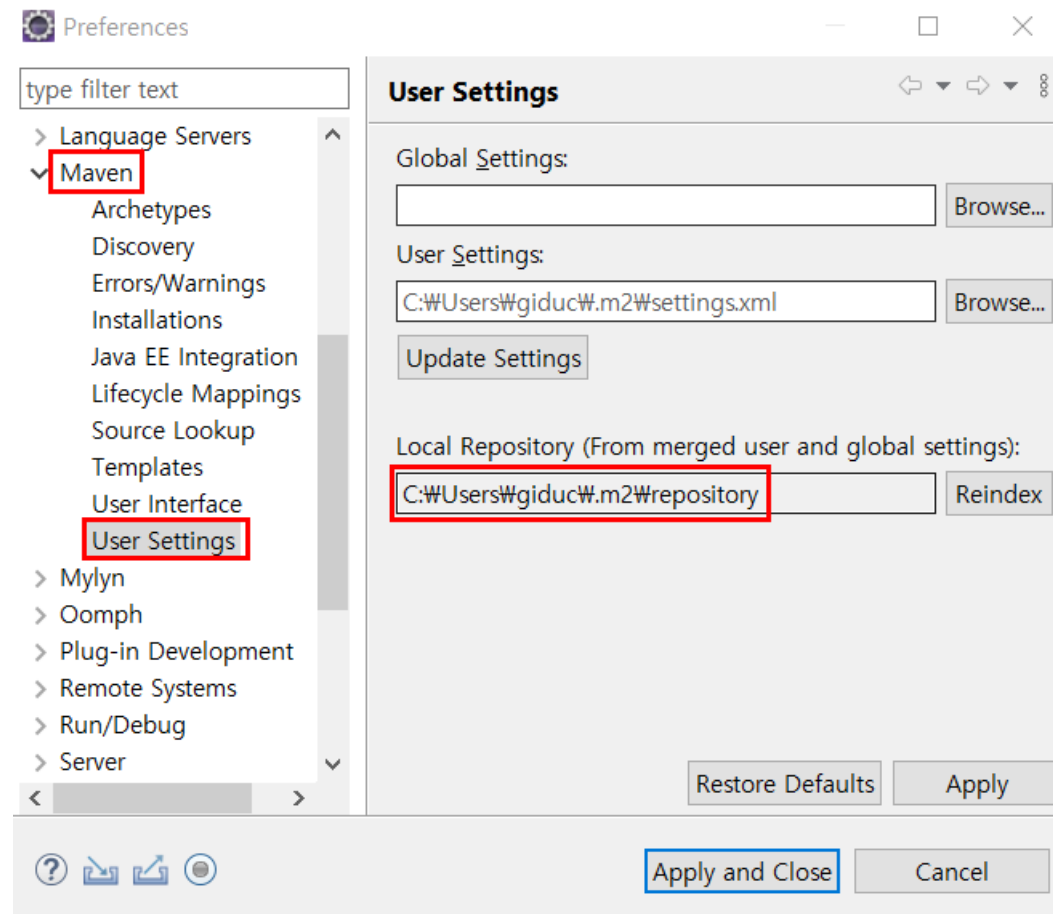
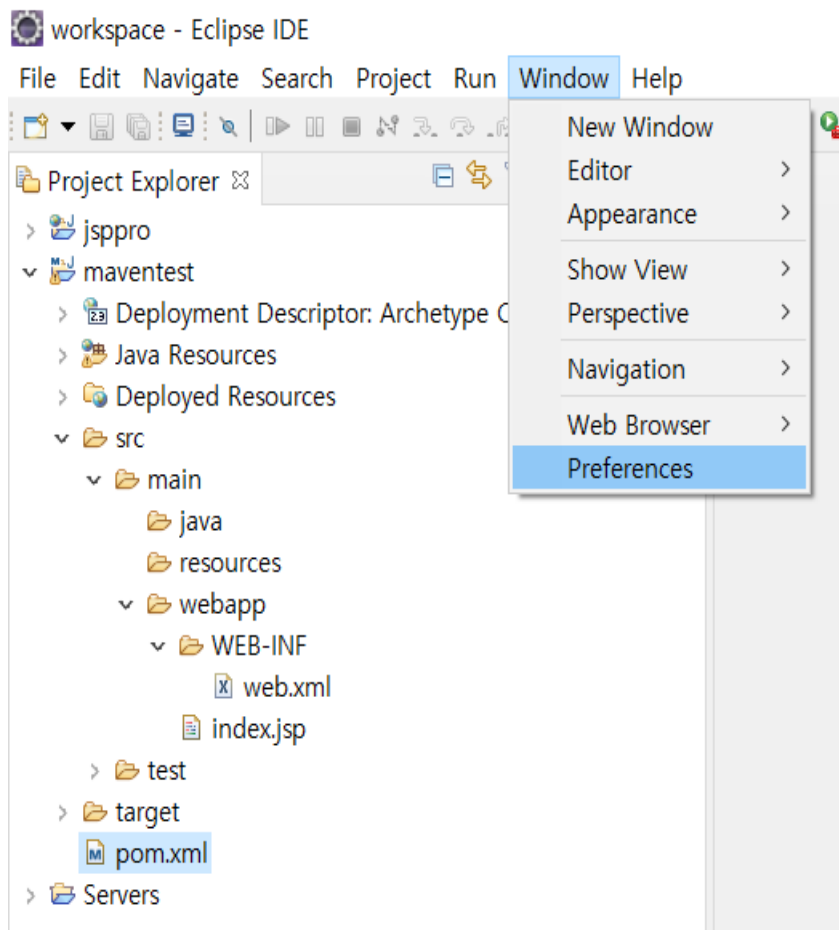
```
        <version>1.2</version>
```

```
    </dependency>
```

```
</dependencies>
```

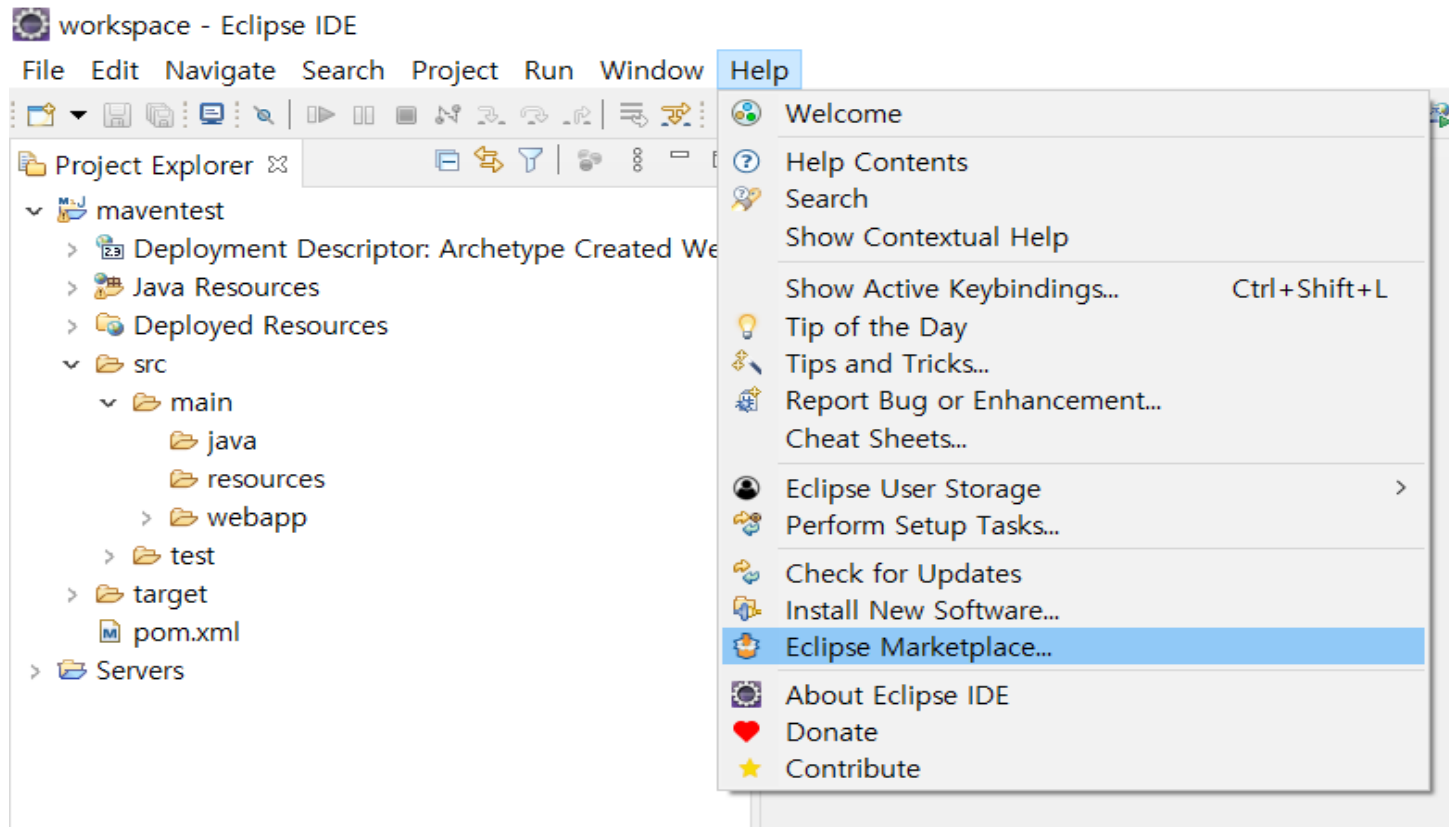
# Local Repository

❖ 로컬 저장소 위치 : `C:\Users\계정\m2\repository`



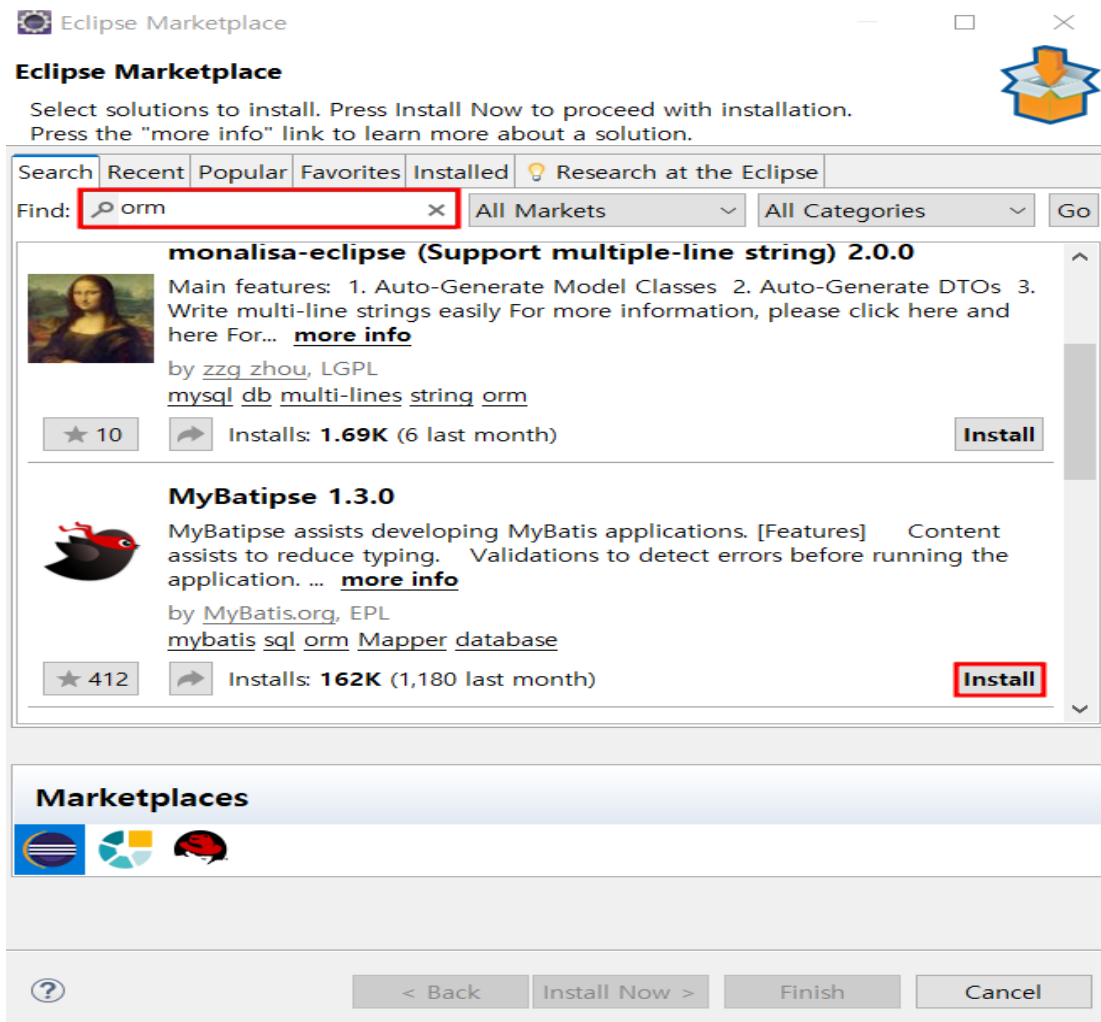
# Eclipse plugin 설치

## ❖ mybatis plugin 설치



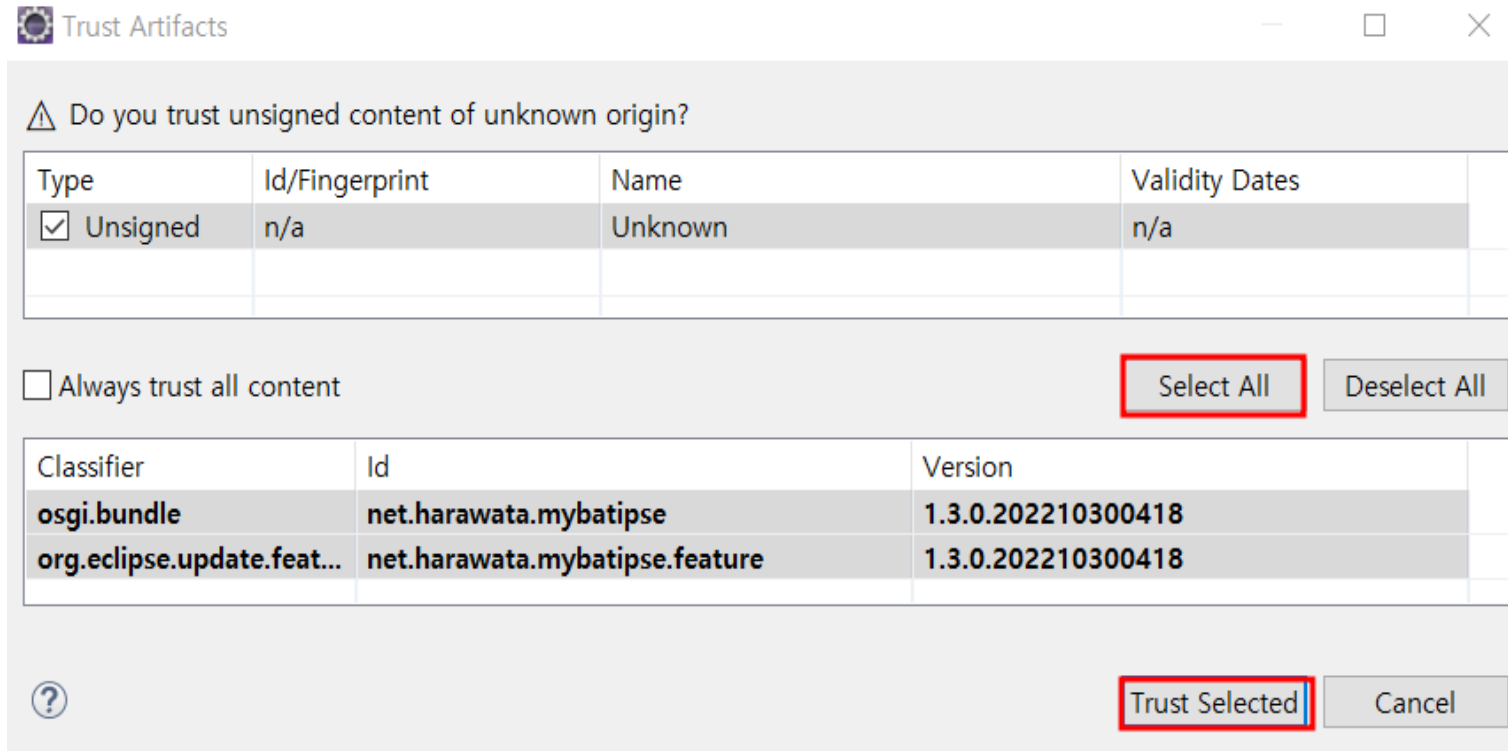
# Eclipse plugin 설치

## ❖ mybatis plugin 설치



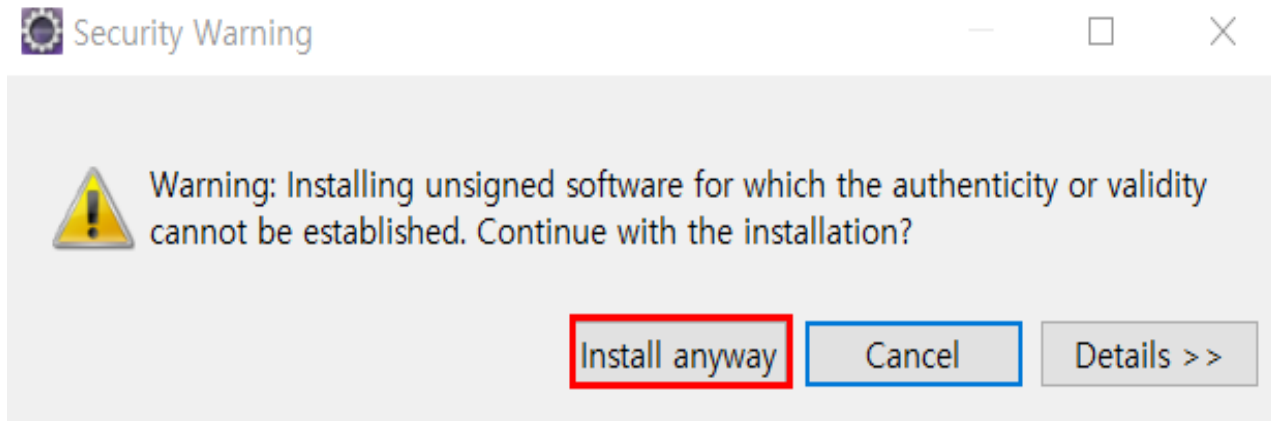
# Eclipse plugin 설치

## ❖ mybatis plugin 설치



# Eclipse plugin 설치

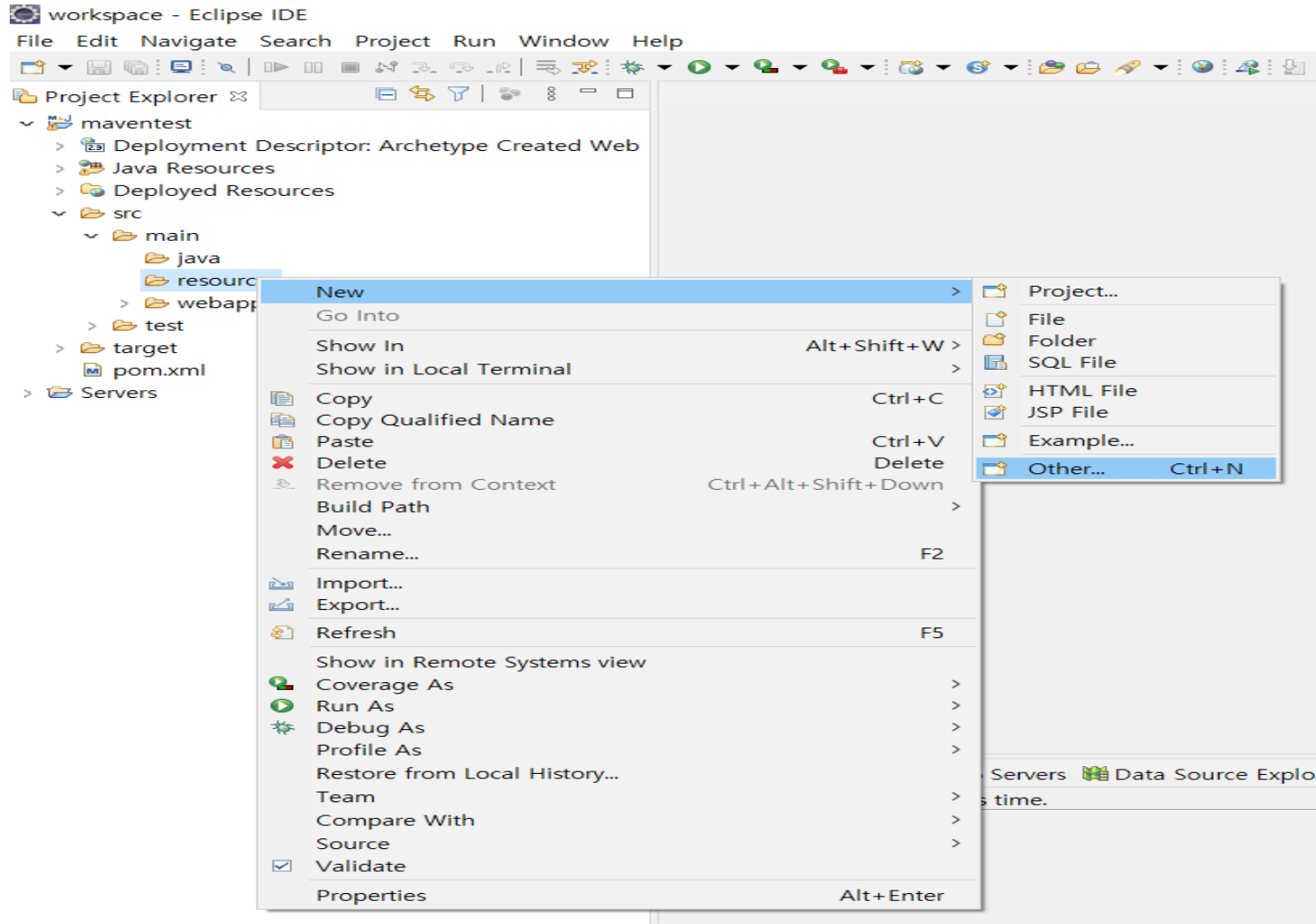
❖ mybatis plugin 설치





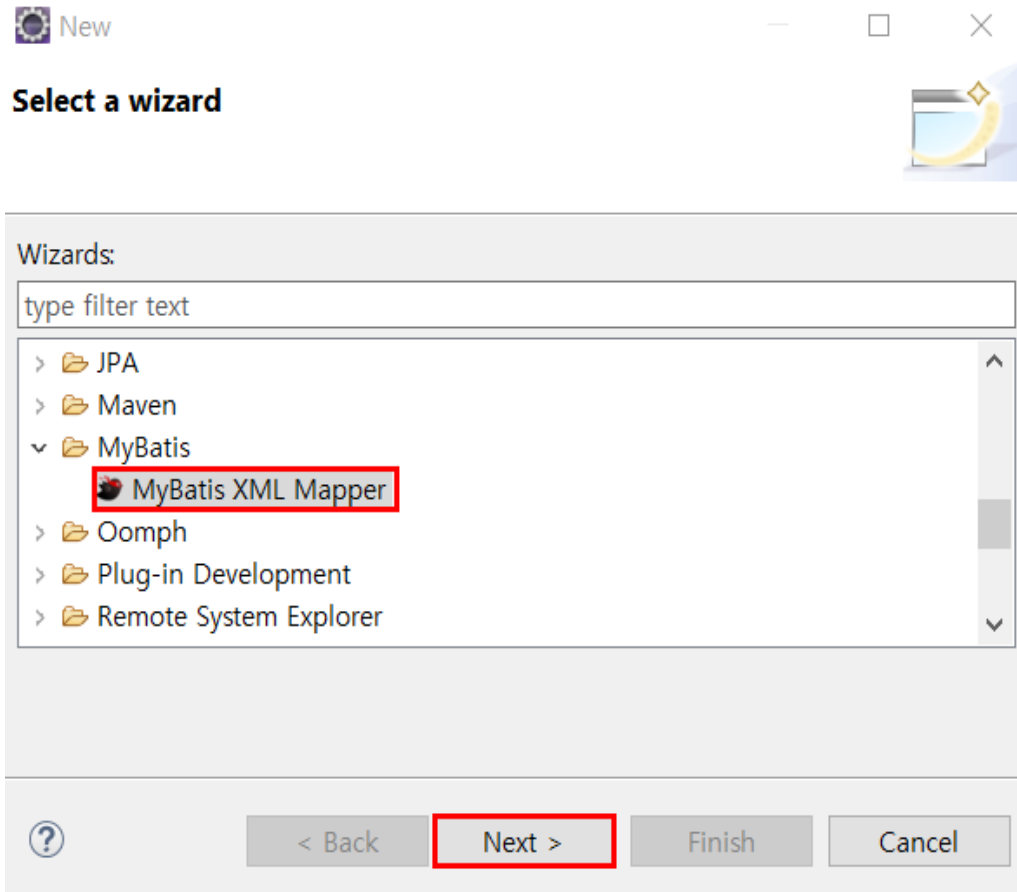
# Mabatis Mapper

## ❖ mybatis mapper 파일 생성



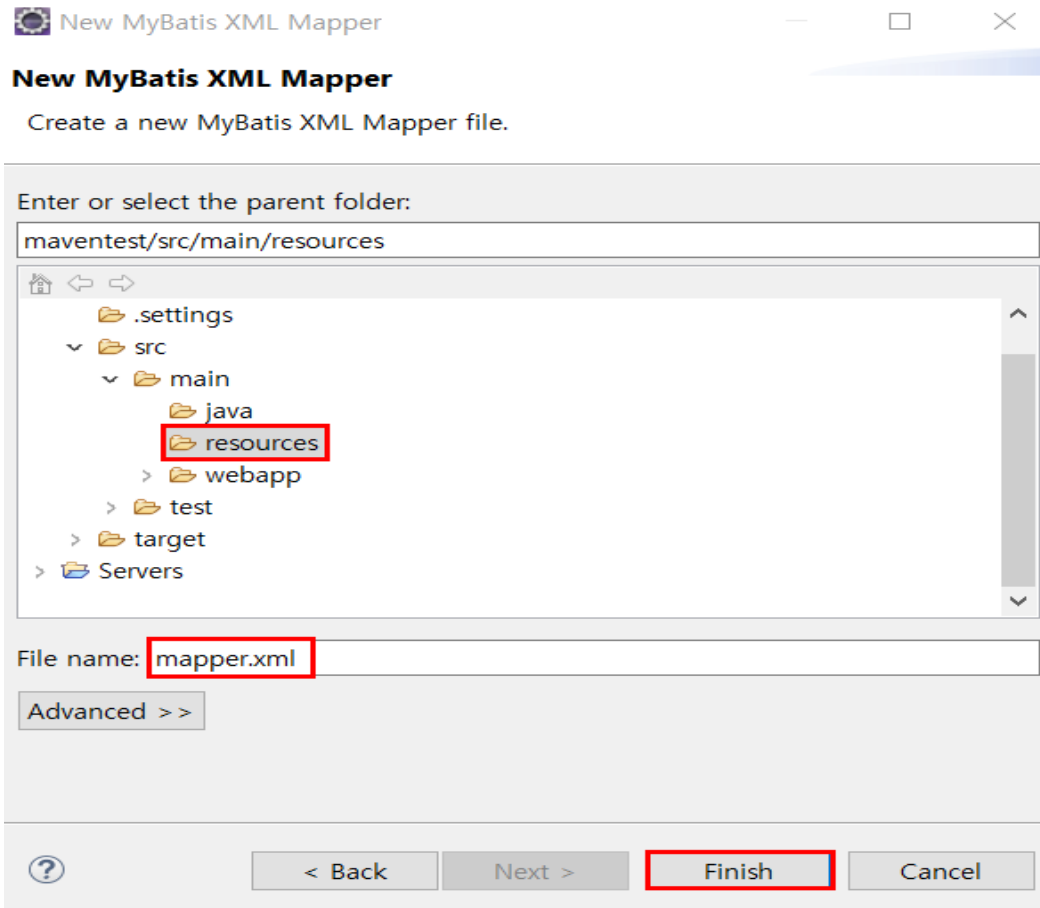
# Mabatis Mapper

## ❖ mybatis mapper 파일 생성



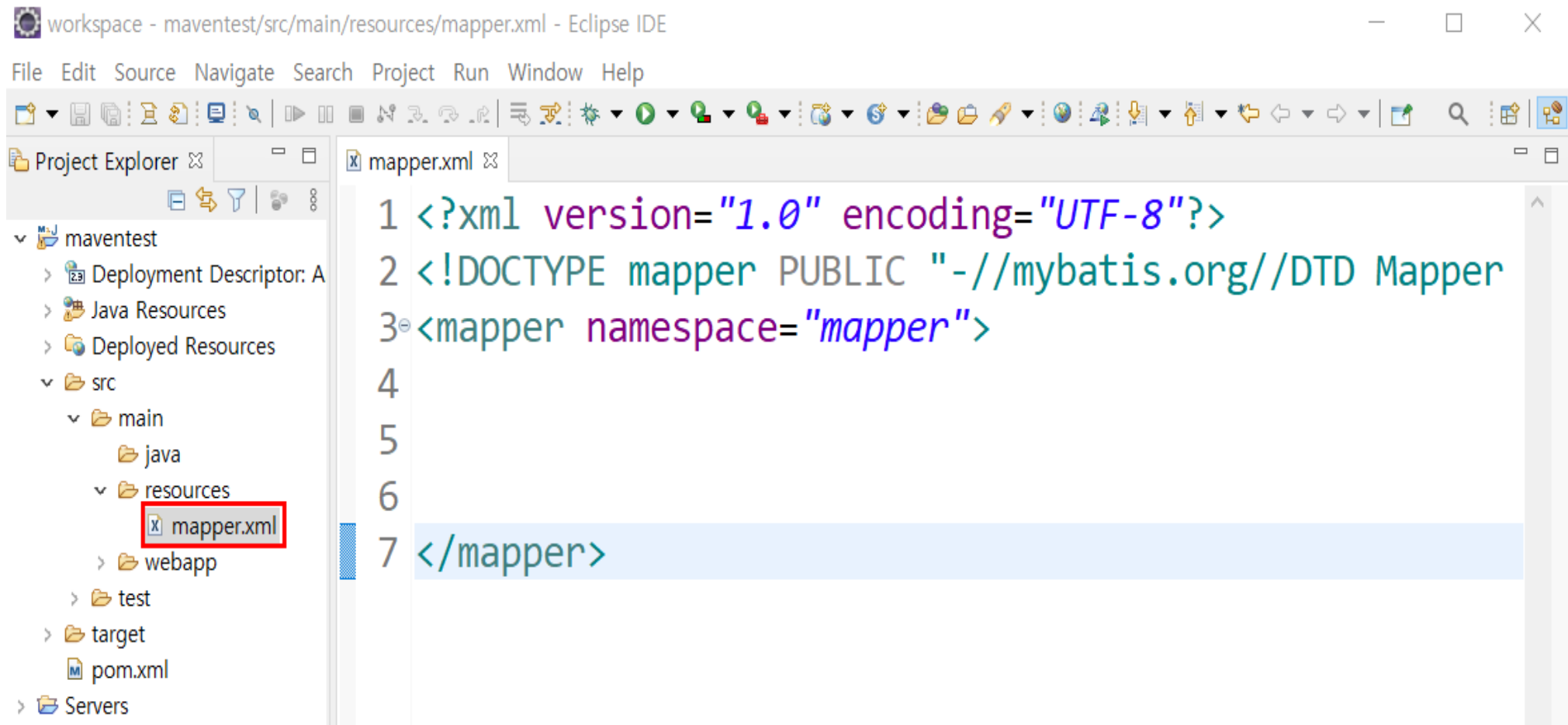
# Mabatis Mapper

## ❖ mybatis mapper 파일 생성



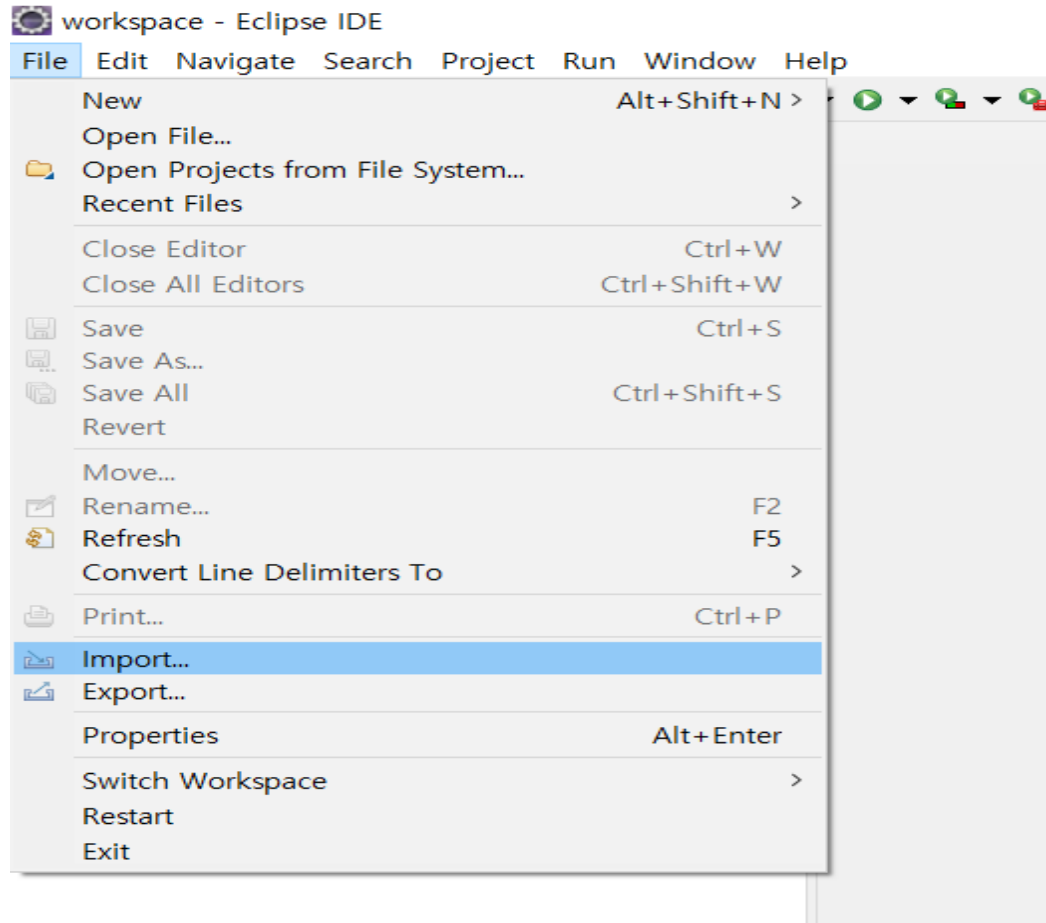
# Mabatis Mapper

## ❖ mybatis mapper 파일 생성



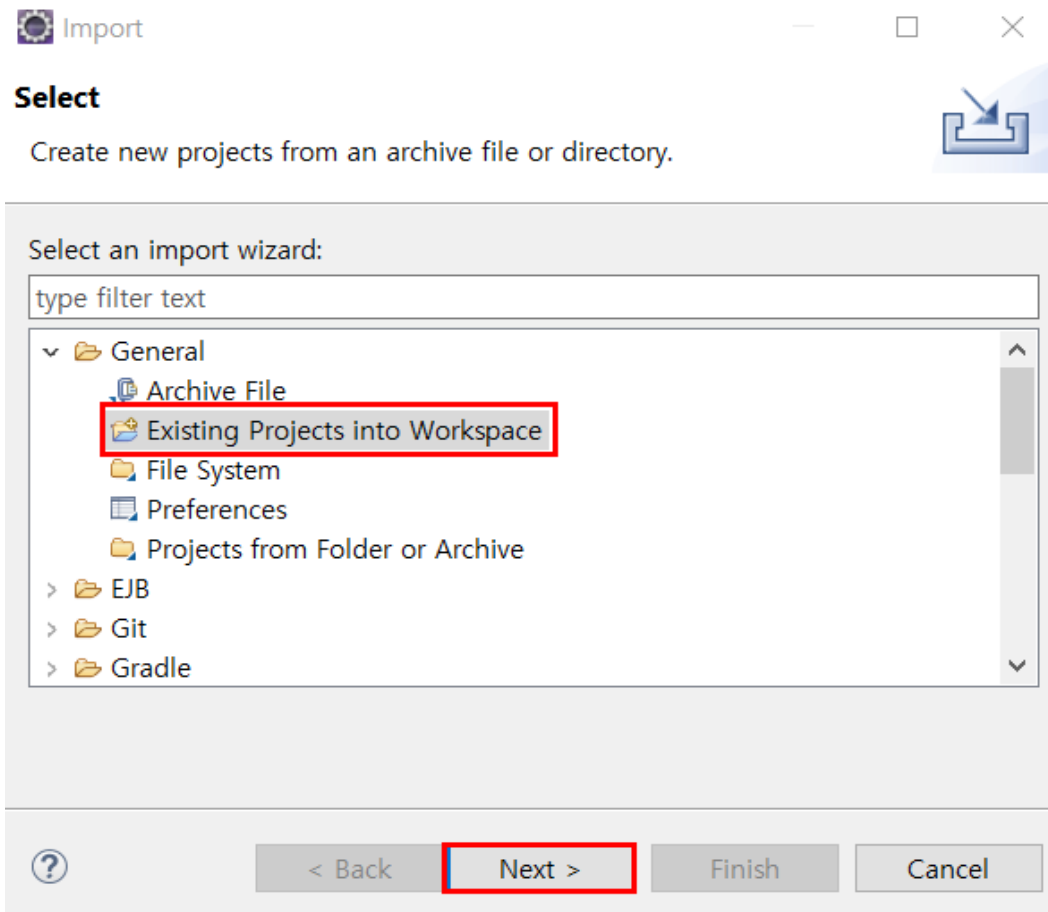
# mybatistest project

## ❖ mybatistest project import



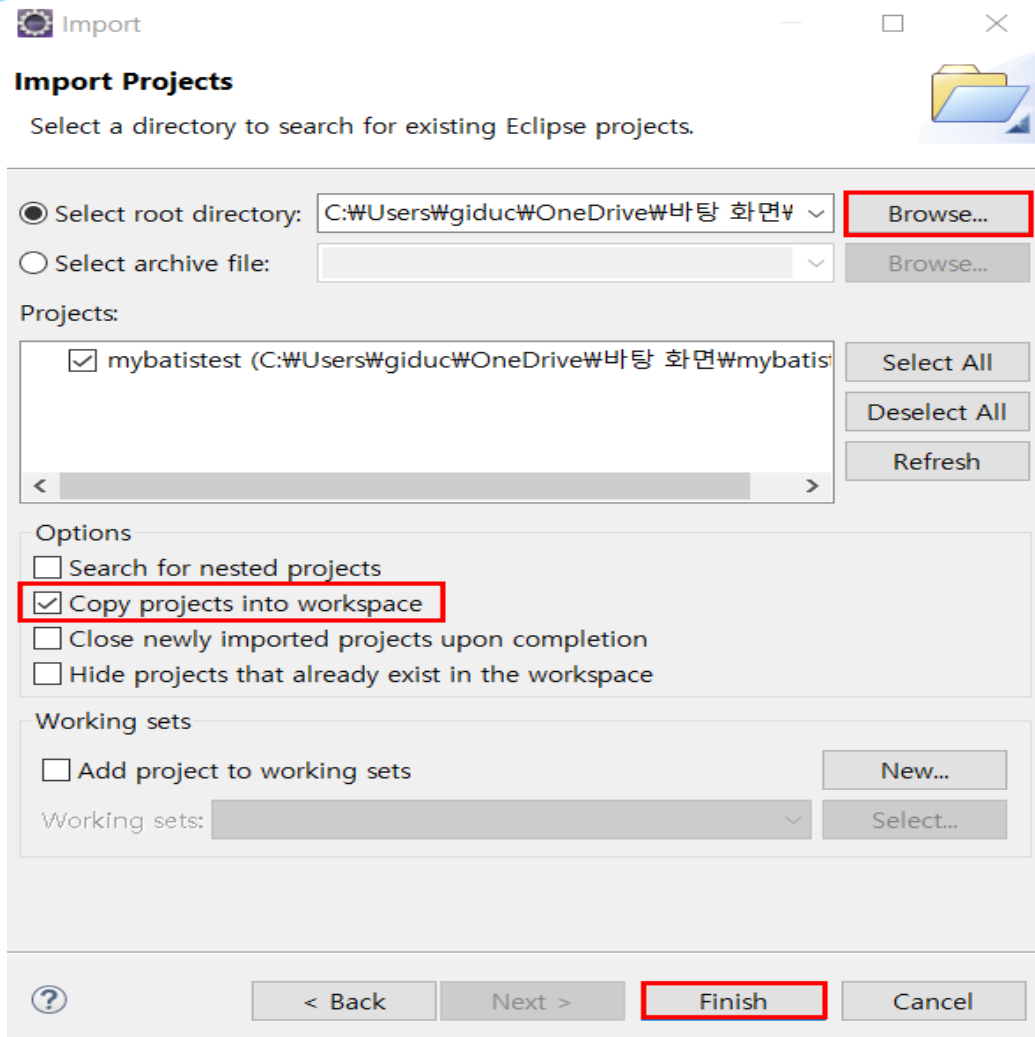
# mybatistest project

## ❖ mybatistest project import



# mybatistest project

## ❖ mybatistest project import



# mybatistest project

## ❖ mybatistest project import

- ▼ mybatistest
  - > Deployment Descriptor: Archetype Created Web Application
  - > Java Resources
  - > Deployed Resources
  - ▼ src
    - ▼ main
      - java
      - ▼ resources
        - db.properties
        - Mapper.xml
        - mybatis-config.xml
      - > webapp
    - > test
  - > target
  - pom.xml



# mybatistest project

## ❖ db.properties

#mysql

#jdbc.driverClassName=com.mysql.jdbc.Driver

#jdbc.url=jdbc:mysql://localhost:3306/jsptest

#jdbc.username=jspid

#jdbc.password=jsppass

#oracle

jdbc.driverClassName=oracle.jdbc.driver.OracleDriver

jdbc.url=jdbc:oracle:thin:@localhost:1521:xe

jdbc.username=scott

jdbc.password=tiger

# mybatistest project

❖ **mybatis-config.xml** : mybatis 환경 설정 파일

```
<configuration>
  <properties resource= "db.properties" />
  <typeAliases>
    <typeAlias type= "#package.#modelname" alias= "#modelname"></typeAlias>
  </typeAliases>
  <environments default= "development">
    <environment id= "development">
      <transactionManager type= "JDBC" />
      <dataSource type= "POOLED">
        <property name= "driver" value= "${jdbc.driverClassName}" />
        <property name= "url" value= "${jdbc.url}" />
        <property name= "username" value= "${jdbc.username}" />
        <property name= "password" value= "${jdbc.password}" />
      </dataSource>
    </environment>
  </environments>
  <mappers>
    <mapper resource= "#package/#mapper.xml" />
  </mappers>
</configuration>
```

# mybatistest project

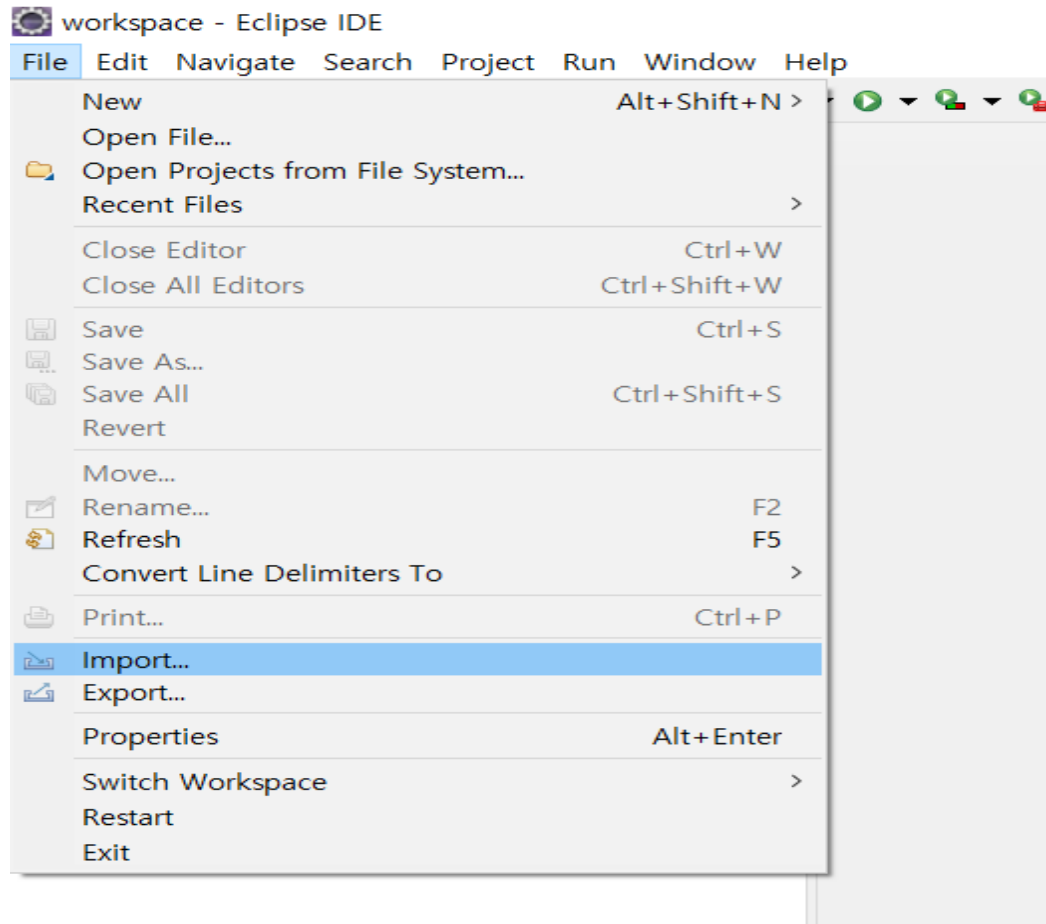
❖ **Mapper.xml** : mapper 파일

```
<mapper namespace= "#package.#mapperinterfacename">
  <select id= "getSomething" parameterType= "int" resultType= "#package.#modelname">
    SELECT
    columnname1,
    columnname2,
    columnname3
    FROM tablename1
    WHERE columnname1 = #{value}
  </select>
  <resultMap type= "#modelname" id= "YourResultSet">
    <id property= "param1" column= "columnname1" />
    <result property= "param2" column= "columnname2" />
    <result property= "param3" column= "columnname3" />
  </resultMap>
  <select id= "getAll" resultMap= "YourResultSet">
    SELECT * FROM tablename1
  </select>
  <insert id= "insertSomething" parameterType= "#modelname">
    INSERT INTO tablename1 (columnname1, columnname2, columnname3)
    VALUES(#{value1},#{value2},#{value3})
  </insert>
  <update id= "updateSomething" parameterType= "#modelname">
    UPDATE tablename1
    SET
    columnname1= #{param1}
    WHERE
    columnname2 = #{param2}
  </update>
  <delete id= "deleteSomething" parameterType= "int">
    DELETE FROM tablename1 WHERE
    columnname1 = #{param1}
  </delete>
</mapper>
```

# JSP-MyBatis 연동

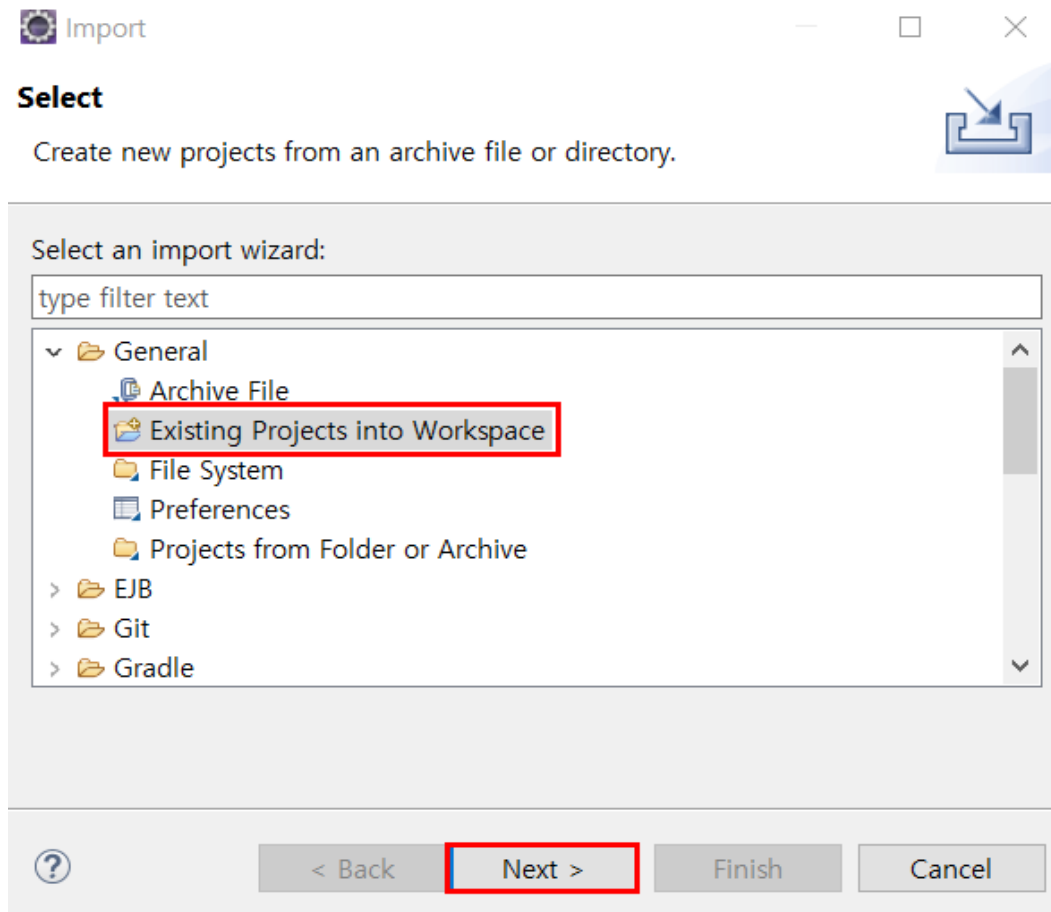
# member22 project

## ❖ member22 project import



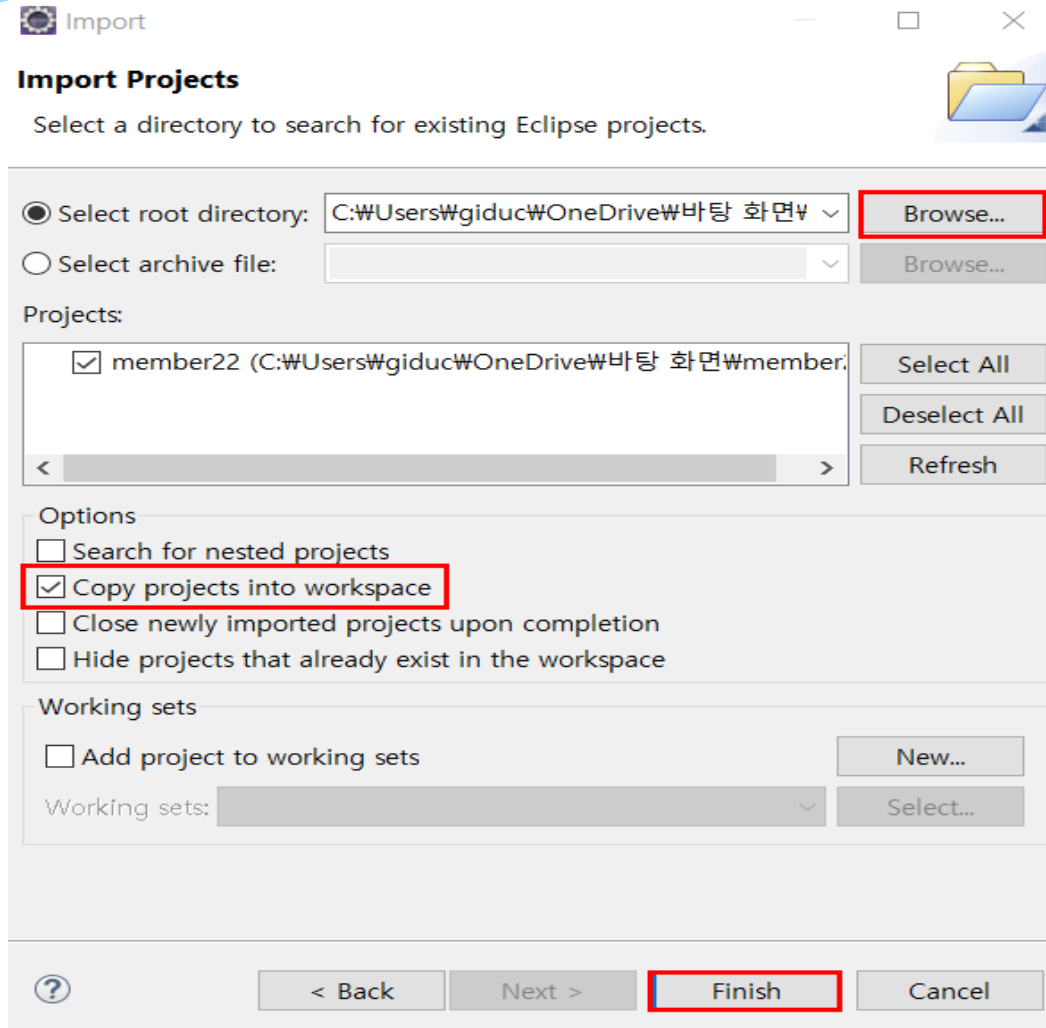
# member22 project

## ❖ member22 project import



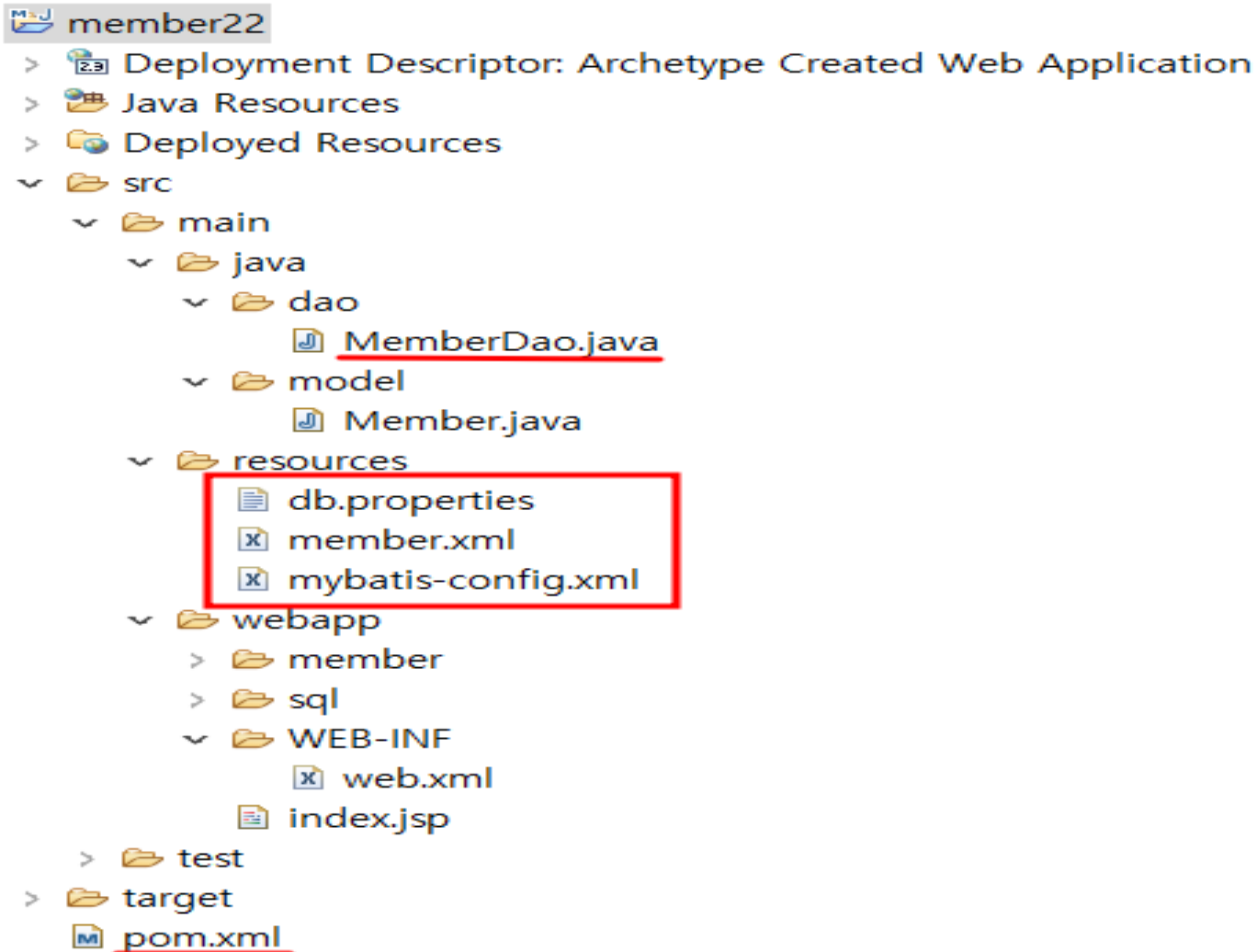
# member22 project

## ❖ member22 project import



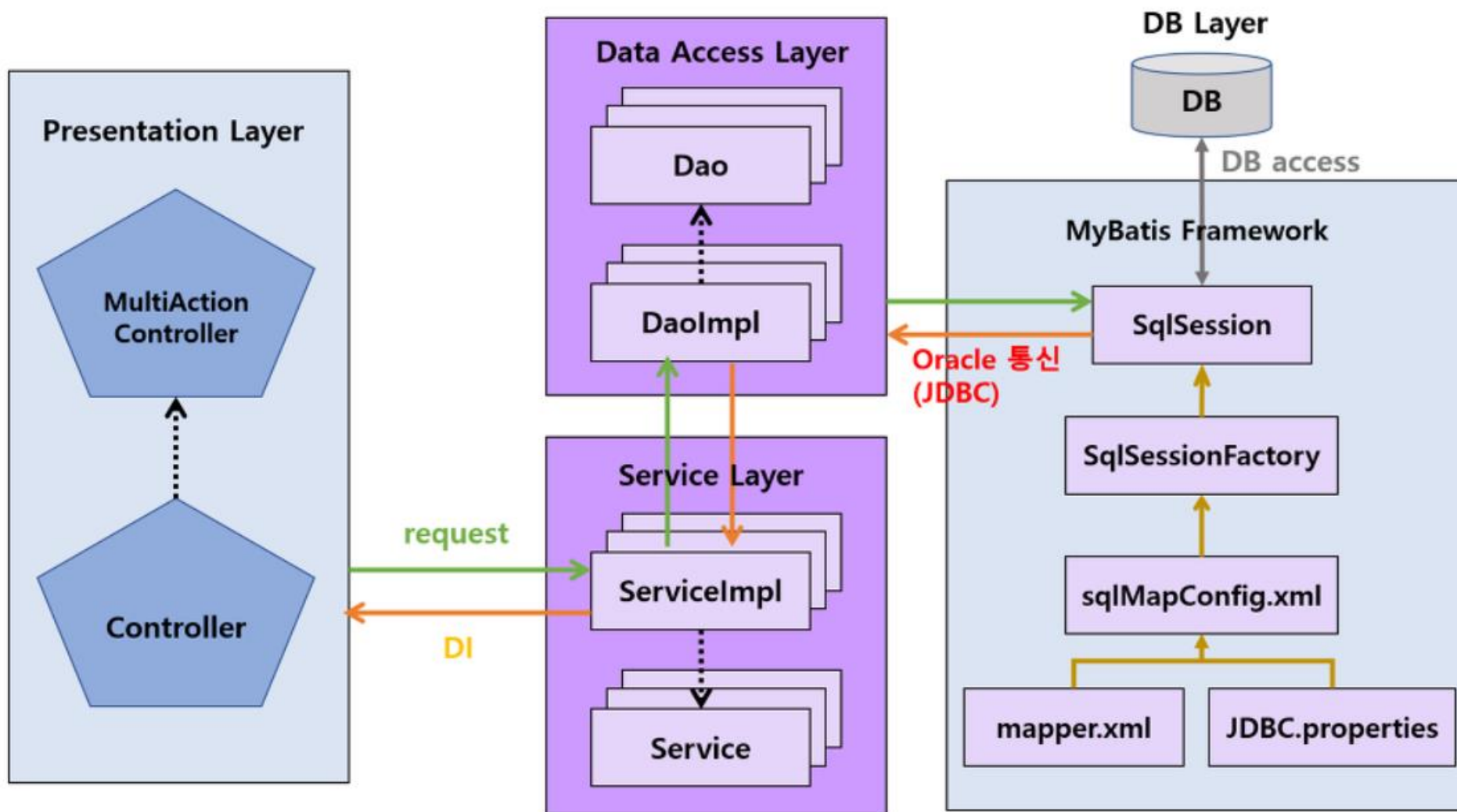
# member22 project

## ❖ member22 project





# Mybatis 데이터 액세스 계층 흐름도



# Mybatis 데이터 액세스 계층 흐름도



# MemberDao.java (1/7)

```
public class MemberDao {  
  
    private SqlSession getSession() {  
        SqlSession session=null;  
        Reader reader=null;  
        try {  
            reader = Resources.getResourceAsReader("mybatis-config.xml");  
            SqlSessionFactory sf = new SqlSessionFactoryBuilder().build(reader);  
            session = sf.openSession(true);           // auto commit  
        } catch (IOException e) {  
            System.out.println(e.getMessage());  
        }  
        return session;  
    }  
}
```

# MemberDao.java (2/7)

```
public int insert(Member member) {  
    int result = 0;  
    SqlSession session=null;  
    try {  
        session = getSession();  
        result = session.insert("insert", member);  
    } catch (Exception e) {  
        System.out.println(e.getMessage());  
    }  
    return result;  
}
```

# MemberDao.java (3/7)

```
public int chk(Member member) {  
    int result = 0;  
    SqlSession session=null;  
    try { session = getSession();  
        Member mem = (Member) session.selectOne("select", member.getId());  
        if (mem.getId().equals(member.getId())) {  
            result = -1;  
            if (mem.getPassword().equals(member.getPassword())) {  
                result = 1;  
            }  
        }  
    } catch (Exception e) {  
        System.out.println(e.getMessage());  
    }  
    return result;  
}
```

# MemberDao.java (4/7)

```
public Member select(String id) throws SQLException {  
    Member mem = null;  
    SqlSession session=null;  
    try { session = getSession();  
        mem = (Member) session.selectOne("select", id);  
    } catch (Exception e) {  
        System.out.println(e.getMessage());  
    }  
    return mem;  
}
```

# MemberDao.java (5/7)

```
public List<Member> list() {  
    List<Member> list = new ArrayList<Member>();  
    SqlSession session=null;  
    try { session = getSession();  
        list = session.selectList("list");  
    } catch (Exception e) {  
        System.out.println(e.getMessage());  
    }  
    return list;  
}
```



# MemberDao.java (6/7)

```
public int delete(String id) {  
    int result = 0;  
    SqlSession session=null;  
    try { session = getSession();  
        result = session.delete("delete", id);  
    } catch (Exception e) {  
        System.out.println(e.getMessage());  
    }  
    return result;  
}
```



# MemberDao.java (7/7)

```
public int update(Member mem) {  
    int result = 0;  
    SqlSession session=null;  
    try { session = getSession();  
        result = session.update("update", mem);  
    } catch (Exception e) {  
        System.out.println(e.getMessage());  
    }  
    return result;  
}
```

# mybatis-config.xml

```
<configuration>
  <properties resource="db.properties" />
  <typeAliases>
    <typeAlias type="model.Member" alias="member"></typeAlias>
  </typeAliases>
  <environments default="development">
    <environment id="development">
      <transactionManager type="JDBC" />
      <dataSource type="POOLED">
        <property name="driver" value="${jdbc.driverClassName}" />
        <property name="url" value="${jdbc.url}" />
        <property name="username" value="${jdbc.username}" />
        <property name="password" value="${jdbc.password}" />
      </dataSource>
    </environment>
  </environments>
  <mappers>
    <mapper resource="member.xml" />
  </mappers>
</configuration>
```

# db.properties

#oracle

jdbc.driverClassName=oracle.jdbc.driver.OracleDriver

jdbc.url=jdbc:oracle:thin:@localhost:1521:xe

jdbc.username=totoro

jdbc.password=totoro123

# member.xml

```
<mapper namespace="members">

    <insert id="insert" parameterType="member">
        insert into member22 values (#{id}, #{password})
    </insert>

    <select id="List" resultType="member">
        select * from member22
    </select>

    <select id="select" parameterType="String" resultType="member">
        select * from member22 where id = #{id}
    </select>

    <update id="update" parameterType="member">
        update member22 set password = #{password} where id = #{id}
    </update>

    <delete id="delete" parameterType="String">
        delete from member22 where id = #{id}
    </delete>

</mapper>
```