

# 삼성 청년 SW 아카데미

Front-End Programming

## 학습목표

- 웹프로젝트의 구성계층에 대해 설명한다.
- MyBatis Framework의 사용방법을 익혀서 응용한다.

## 스프링 프레임워크란?

자바 엔터프라이즈 개발을 위한 오픈소스 경량 애플리케이션 프레임워크다.  
공통 프로그래밍 모델 및 Configuration 모델을 제공한다.



프레임워크가 애플리케이션 수준의 인프라 구조를 제공  
개발자가 귀찮은 일에 신경 쓰지 않고 비즈니스 로직 개발에 전념

# JAVA 온라인 라이브 강의

## 웹 프로젝트 3 레이어(tier)

**Confidential**

Presentation  
Layer

UI담당 구성요소

Business Logic  
Layer

서비스계층  
고객 요구사항반영

Data Access  
Layer

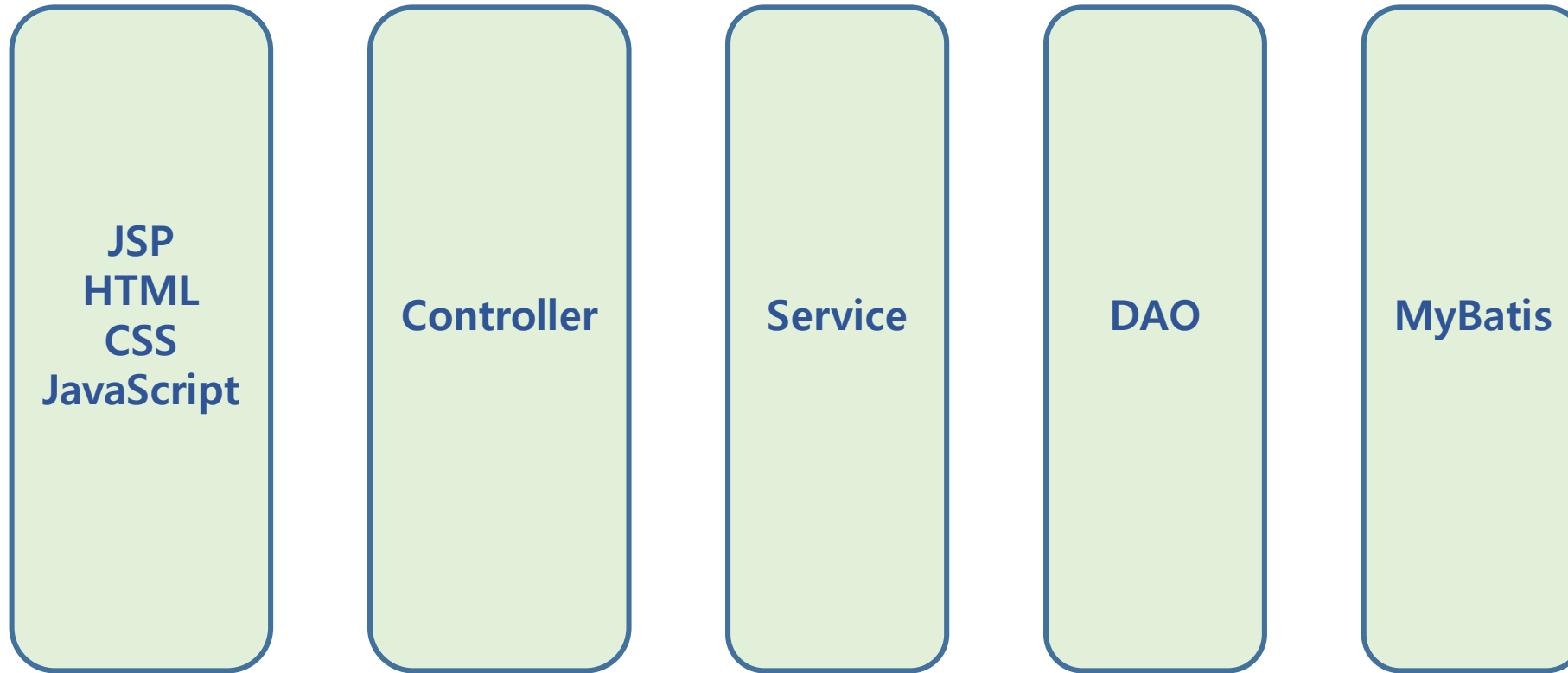
Persistence계층  
데이터처리



# JAVA 온라인 라이브 강의

## 세분화된 프로젝트 구성

**Confidential**



Service주입

DAO주입

SessionSession주입



## MyBatis

- 개발자가 지정한 SQL, 저장프로시저 그리고 몇가지 고급 매핑을 지원하는 퍼시스턴스 프레임워크
- JDBC코드와 수동으로 셋팅하는 파라미터와 결과 매핑을 제거
- 데이터베이스 레코드에 원시타입과 Map인터페이스 그리고 자바 POJO를 설정하고 매핑하기 위해 XML과 애노테이션을 사용

# JAVA 온라인 라이브 강의

Confidential

```
@Override
public MemberDto login(String userid, String userpwd) throws SQLException {
    MemberDto memberDto = null;
    Connection conn = null;
    PreparedStatement pstmt = null;
    ResultSet rs = null;
    try {
        conn = DBUtil.getConnection();
        StringBuilder sql = new StringBuilder();
        sql.append("select username, userid, email \n");
        sql.append("from ssafy_member \n");
        sql.append("where userid = ? and userpwd = ?");
        pstmt = conn.prepareStatement(sql.toString());
        pstmt.setString(1, userid);
        pstmt.setString(2, userpwd);
        rs = pstmt.executeQuery();
        if(rs.next()) {
            memberDto = new MemberDto();
            memberDto.setUserId(rs.getString("userid"));
            memberDto.setUsername(rs.getString("username"));
            memberDto.setEmail(rs.getString("email"));
        }
    } catch (SQLException e) {
        e.printStackTrace();
        memberDto = null;
    } finally {
        DBUtil.close(rs);
        DBUtil.close(pstmt);
        DBUtil.close(conn);
    }
    return memberDto;
}
```

MyBatis

```
<select id="selectLogin">
    select username, userid, email
    from ssafy_member
    where userid = #{userid} and userpwd = #{userpwd}
</select>
```

# JAVA 온라인 라이브 강의

**Confidential**

New Spring Legacy Project

**Spring Legacy Project**

Click 'Next' to load the template contents.

Project name:


☒ Use default location

Location:

Select Spring version:

Templates:

- Simple Spring Web Maven
  - Batch
  - GemFire
  - Integration
  - Persistence
- Simple Spring Utility Project
- Spring MVC Project**

 requires downloading [Configure templates...](#)

Description:  
A new Spring MVC web application development project

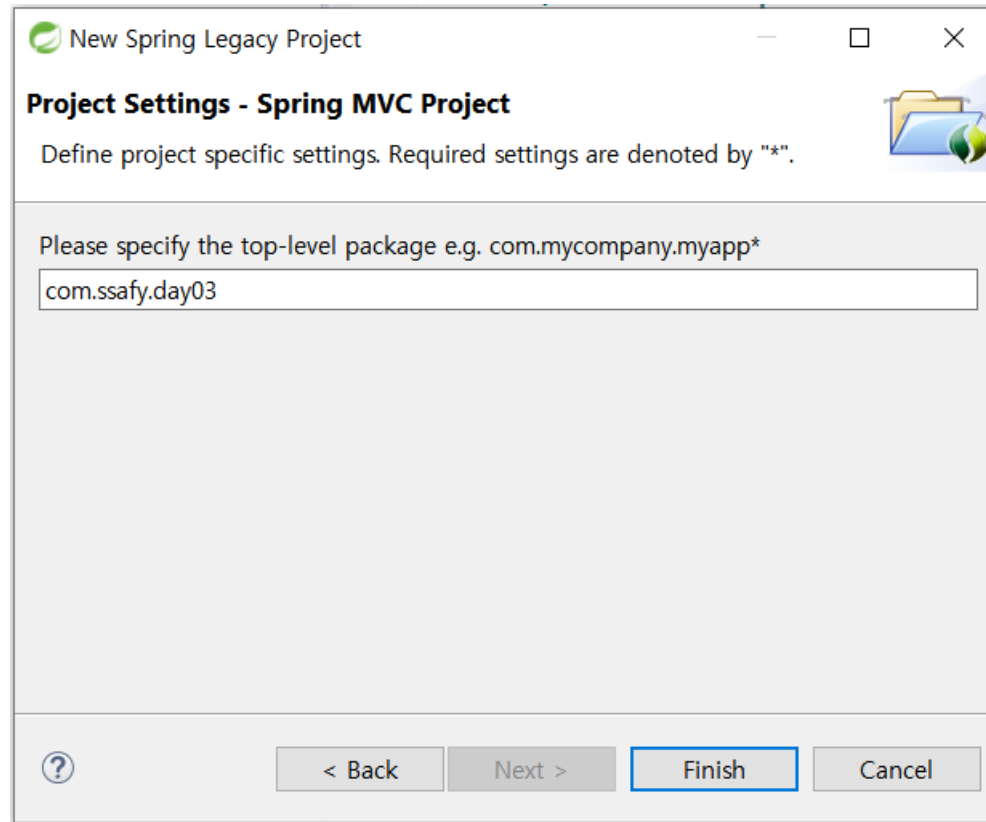
URL: <https://dist.springsource.com/release/STS/help/org.springframework.templates.mvc-3.2.2.zip>

Working sets

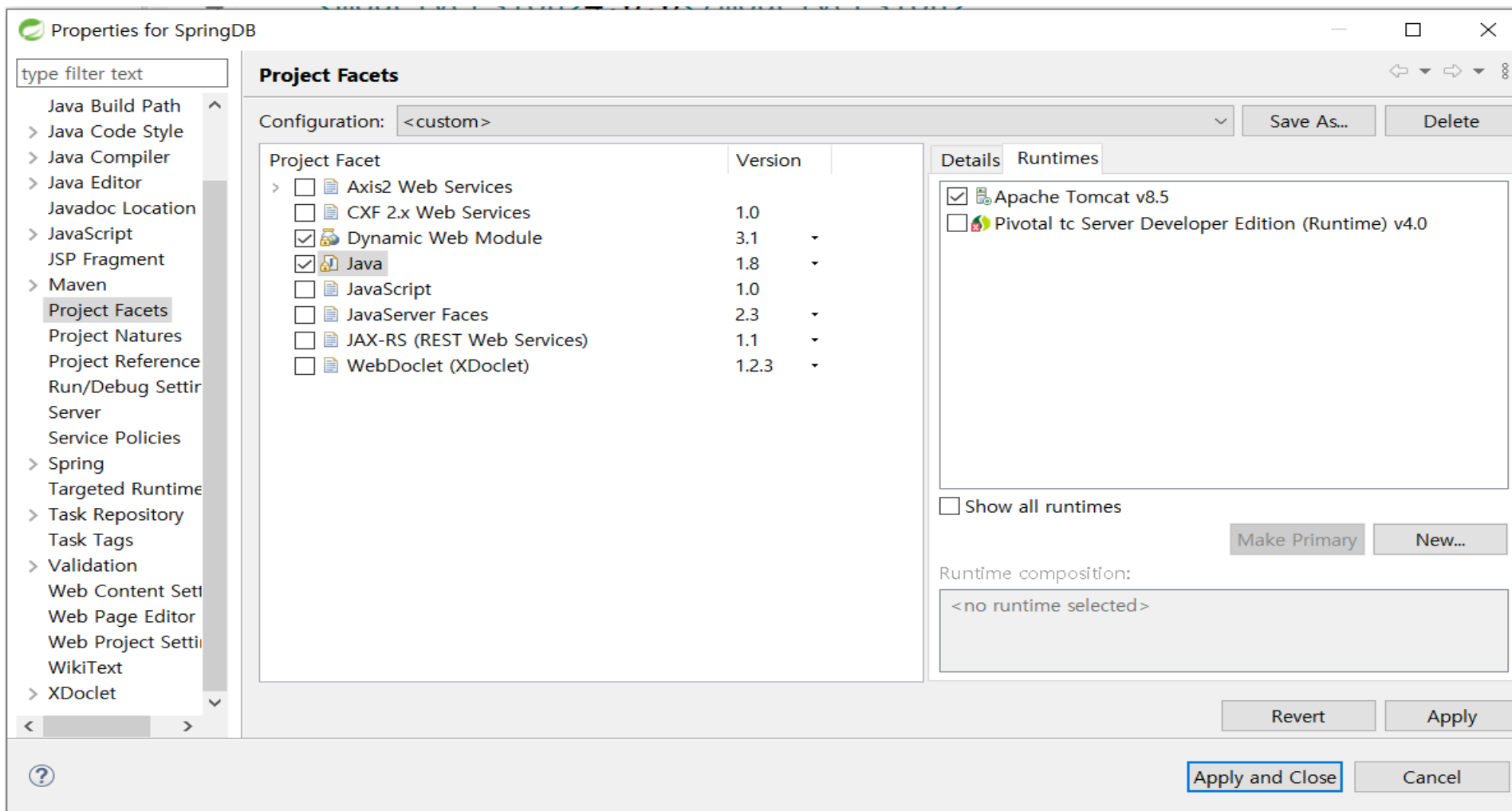
☐ Add project to working sets

Working sets:



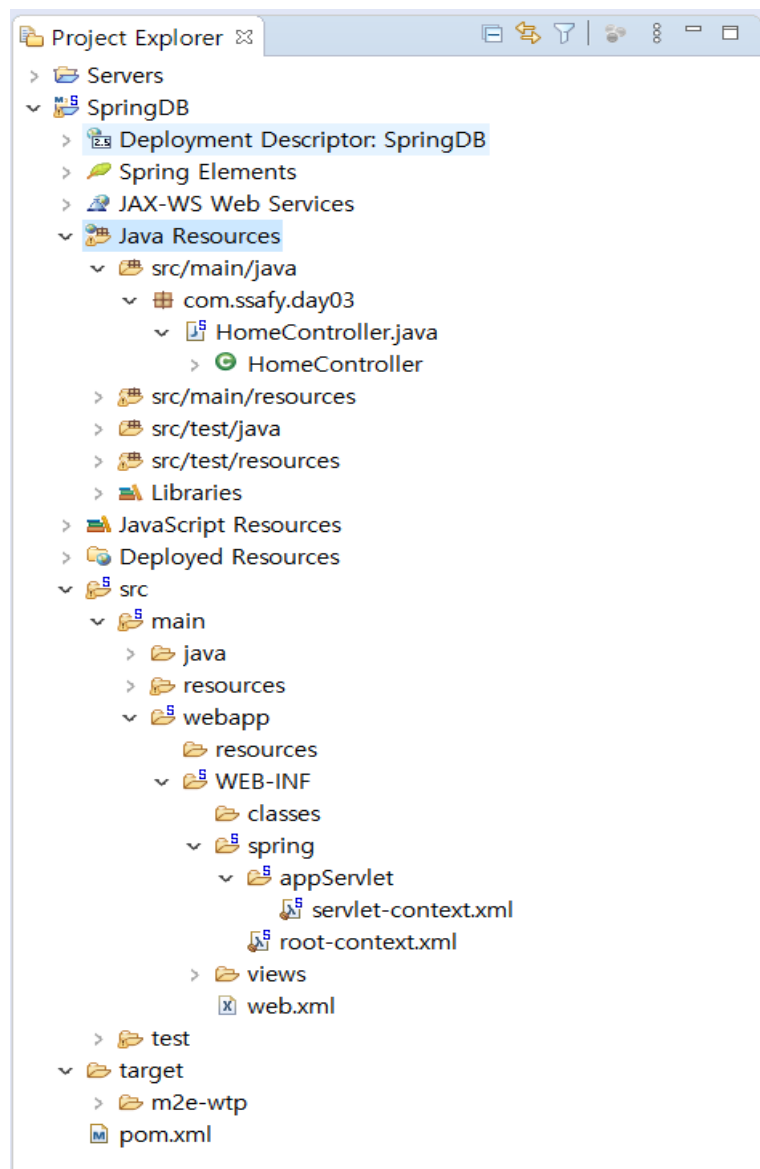


```
SpringDB/pom.xml
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/ma
4   <modelVersion>4.0.0</modelVersion>
5   <groupId>com.ssafy</groupId>
6   <artifactId>day03</artifactId>
7   <name>SpringDB</name>
8   <packaging>war</packaging>
9   <version>1.0.0-BUILD-SNAPSHOT</version>
10  <properties>
11    <java-version>1.8</java-version>
12    <org.springframework-version>5.2.6.RELEASE</org.springframework-version>
13    <org.aspectj-version>1.6.10</org.aspectj-version>
14    <org.slf4j-version>1.6.6</org.slf4j-version>
15  </properties>
16  <dependencies>
```



# JAVA 온라인 라이브 강의

**Confidential**



### <방명록 기능분석>

1. 방명록 리스트 (최신게시물을 맨위로 해서 보여줌)
2. 방명록 입력 (작성자, 이메일, 전화번호, 비밀번호, 내용)
3. 방명록 수정
4. 방명록 삭제

방명록 JSP(View)

방명록 Java(Model, Controller)

### <방명록 화면목록>

1. 리스트화면 (list.jsp)
2. 입력폼 (inputForm.jsp)
3. 수정폼, 수정/삭제 화면 (editFrom.jsp)

---

입력폼

---

[\[게시물 목록으로\]](#)

작성자	<input type="text"/>
email	<input type="text"/>
전화번호	<input type="text"/>
비밀번호	<input type="password"/>
<div></div>	
<input type="button" value="입력"/> <input type="button" value="취소"/>	

---

방명록: 리스트

---

방명록 쓰기

번호	작성자	전화번호	작성일	내용
3	홍길동	010-1234-5678	2018-07-17	<u>안녕</u>
2	길라임	010-2222-5678	2018-07-17	<u>안녕라임</u>
1	김주원	010-3333-5678	2018-07-17	<u>주원이야</u>

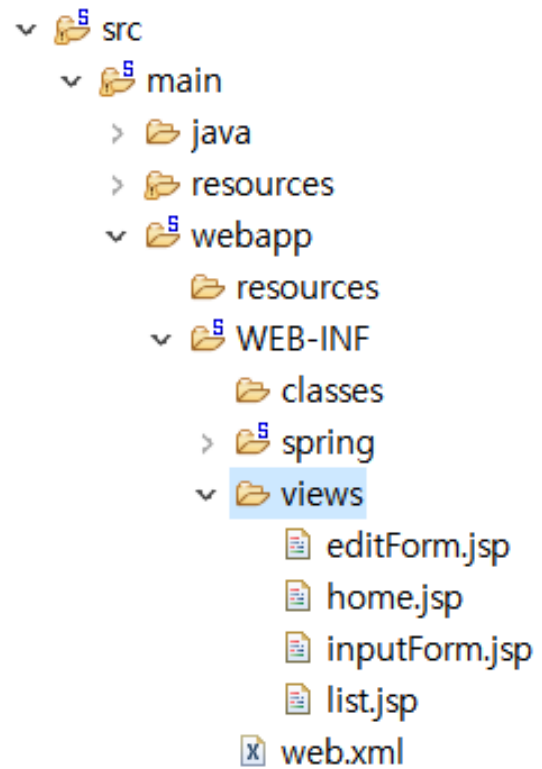
수정폼

[\[게시물 목록으로\]](#)

작성자	<input type="text" value="홍길동"/>
email	<input type="text" value="gildong@daum.net"/>
전화번호	<input type="text" value="010-1234-5678"/>
비밀번호	<input type="password" value="....."/>
<input type="text" value="안녕, 동에번쩍 서에번쩍"/>	
<input type="button" value="수정"/> <input type="button" value="취소"/> <input type="button" value="삭제"/>	








- ▼ Java Resources
  - ▼ src/main/java
    - > com.ssafy.day03
    - ▼ com.ssafy.day3.controller
      - > GuestControlller.java
    - ▼ com.ssafy.day3.dto
      - > Guest.java
    - ▼ com.ssafy.day3.repository
      - > GuestDAO.java
      - > GuestDAOImpl.java
    - ▼ com.ssafy.day3.service
      - > GuestService.java
      - > GuestServiceImpl.java



```
Guest.java
1 package com.ssafy.day3.dto;
2
3 import java.util.Date;
4
5 public class Guest {
6     private int no;
7     private String writer;
8     private String email;
9     private String tel;
10    private String pass;
11    private String contents;
12    private Date wdate;
13
```

```
GuestDAO.java
1 package com.ssafy.day3.repository;
2
3 import com.ssafy.day3.dto.Guest;
4
5 public interface GuestDAO {
6     public void insert(Guest guest);
7     public void update(Guest guest);
8     public void delete(int no);
9     public void select(int no);
10    public void selectAll();
11 }
```

```
GuestService.java ✕
1 package com.ssafy.day3.service;
2
3 import com.ssafy.day3.dto.Guest;
4
5 public interface GuestService {
6     public void registry(Guest guest);
7     public void modify(Guest guest);
8     public void remove(int no);
9     public void find(int no);
10    public void findAll();
11 }
```

- ▼  src/test/java
  - ▼  com.ssafy.day03.db
    - >  DataSourceTest.java
    - >  JDBCTest.java
    - >  MyBatisTest.java

```
package com.ssafy.day03.db;

import java.sql.Connection;

public class JDBCtest {

    @Test
    public void testConnection() throws Exception{
        //1.드라이버 로딩 2.연결객체생성
        Class.forName("com.mysql.cj.jdbc.Driver");
        Connection conn=null;

        try {
            conn= DriverManager.getConnection
("jdbc:mysql://127.0.0.1:3306/ssafyweb?serverTimezone=UTC&useUnicode=yes&characterEncoding=UTF-8",
                "ssafy","ssafy");
            System.out.println("DB연결성공!!");
        }catch(Exception e) {
        }finally {
            if(conn != null) conn.close();
        }
    }
}
//testConnection
```

```
<!-- MySQL 드라이버추가 -->
<dependency>
  <groupId>mysql</groupId>
  <artifactId>mysql-connector-java</artifactId>
  <version>8.0.13</version>
</dependency>
```



```
package com.ssafy.day03.db;

import java.sql.Connection;
import javax.sql.DataSource;

import org.junit.Test;
import org.junit.runner.RunWith;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.test.context.ContextConfiguration;
import org.springframework.test.context.junit4.SpringJUnit4ClassRunner;

@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = {"file:src/main/webapp/WEB-INF/spring/**/*.xml"})
public class DataSourceTest {

    @Autowired
    private DataSource ds; // 기본값 ds=null;

    @Test
    public void testConnection() throws Exception {
        Connection conn = ds.getConnection();
        System.out.println("DBCP DB 연결 성공 ==> " + conn);
    }
}
```

```
<!-- 스프링 JDBC연동 -->  
<dependency>  
  <groupId>org.springframework</groupId>  
  <artifactId>spring-jdbc</artifactId>  
  <version>${org.springframework-version}</version>  
</dependency>
```

```
<bean class="org.springframework.jdbc.datasource.DriverManagerDataSource"
      id="dataSource">
  <property name="driverClassName"
    value="com.mysql.cj.jdbc.Driver"></property>
  <property name="url"
    value="jdbc:mysql://127.0.0.1:3306/ssafyweb?serverTimezone=UTC&";
  <property name="username" value="ssafy"></property>
  <property name="password" value="ssafy"></property>
</bean>
```

```
package com.ssafy.day03.db;

import org.apache.ibatis.session.SqlSession;

@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration(locations = {"file:src/main/webapp/WEB-INF/spring/**/*.xml"})
public class MyBatisTest {

    @Autowired//참조하는 Bean Container에서 일치하는 자료형이 있다면 '주입실행'!!
    private SqlSessionFactory sqlFactory;

    @Autowired
    private SqlSession session;

    //ibatis의 SqlMapClient == mybatis의 SqlSession

    @Test
    public void testFactory() throws Exception{
        //SqlSessionFactory객체 테스트
        System.out.println("sqlFactory==> "+ sqlFactory);
    }

    @Test
    public void testSession() throws Exception{
        //SqlSession(MyBatis에 정의된 sql문 호출하는 객체)객체 테스트
        System.out.println("session==> "+ session);
    }
}
```

```
<!-- MyBatis라이브러리 -->
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis</artifactId>
  <version>3.4.1</version>
</dependency> <!-- mybatis.3.4.1.jar -->


<!-- 스프링 MyBatis연동 -->
<dependency>
  <groupId>org.mybatis</groupId>
  <artifactId>mybatis-spring</artifactId>
  <version>1.3.0</version>
</dependency>





<!-- 스프링 Transaction 처리 -->
<dependency>
  <groupId>org.springframework</groupId>
  <artifactId>spring-tx</artifactId>
  <version>${org.springframework-version}</version>
</dependency>
```

```
<!-- XML내에 작성된 sql문을 호출하는 객체: SqlMapClient(iBatis), SqlSession(MyBatis) -->
<bean class="org.mybatis.spring.SqlSessionFactoryBean"
      id="sqlSessionFactory">
  <property name="dataSource" ref="dataSource"></property>
  <property name="configLocation"
            value="classpath:/mybatis-config.xml"></property>

  <!-- sql문 작성된 mapperXML문서 등록 -->
  <property name="mapperLocations"
            value="classpath:mappers/*.xml"></property>
</bean>

<bean class="org.mybatis.spring.SqlSessionTemplate"
      destroy-method="clearCache">
  <constructor-arg ref="sqlSessionFactory"></constructor-arg>
</bean>
```

▼  src/main/resources

-  mappers
-  META-INF
-  log4j.xml
-  mybatis-config.xml

```
<?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>
  <!-- mybatis-config.xml -->
  <typeAliases>
    <package name="com.ssafy.day3.dto"/>
  </typeAliases>
</configuration>
```





# 내일 방송에서 만나요!

삼성 청년 SW 아카데미