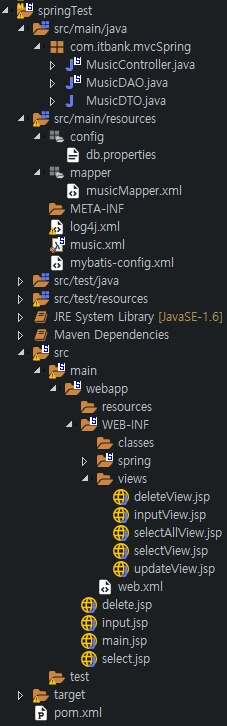
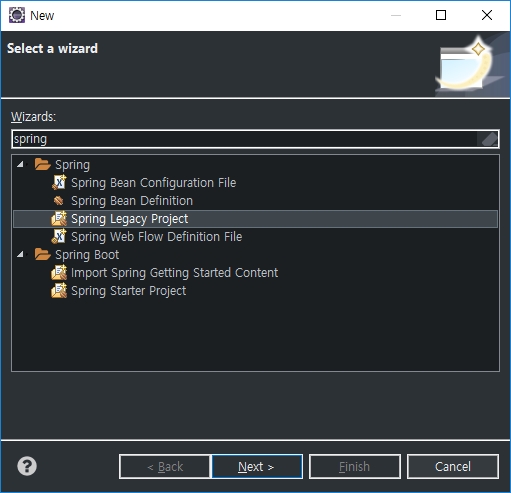
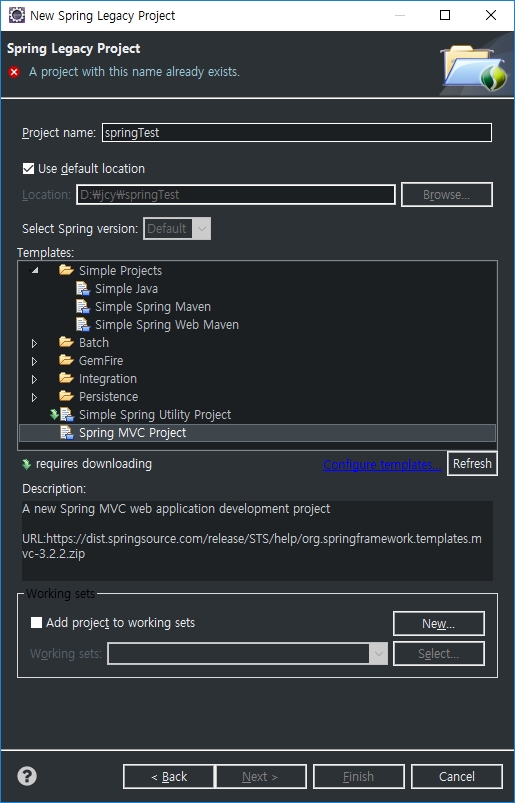
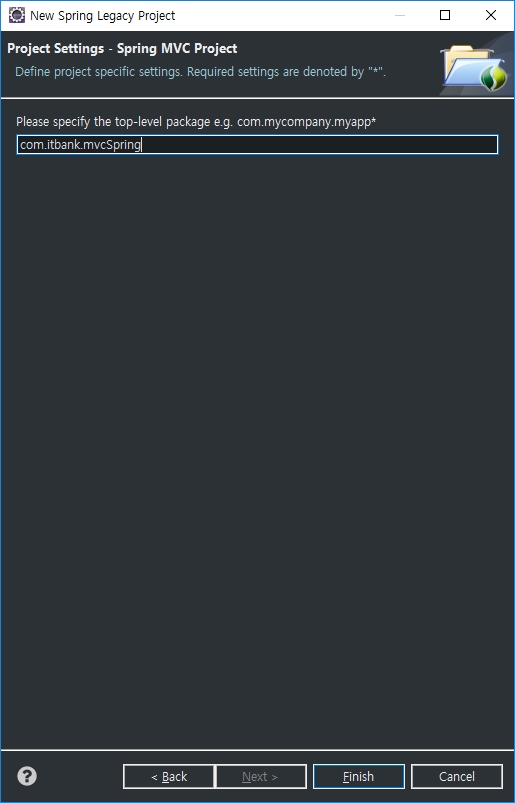
5차시험 - 장채연

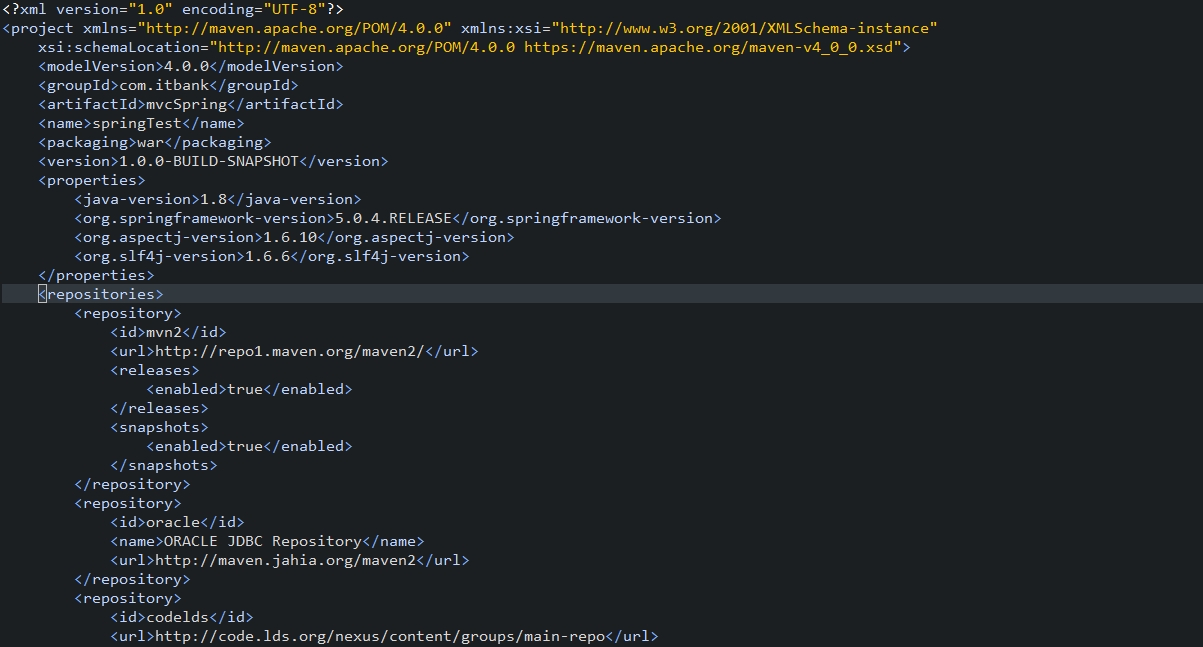
1. 프로젝트 트리



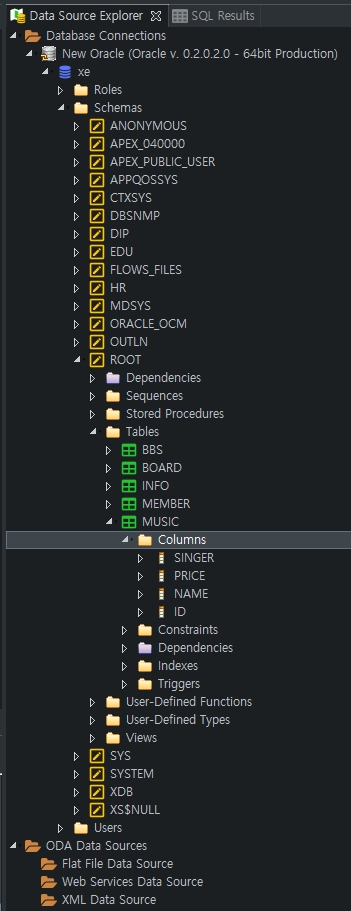
1. 개발환경 구축하기
2. mvc프로젝트를 생성
3. mvc프로젝트 셋팅



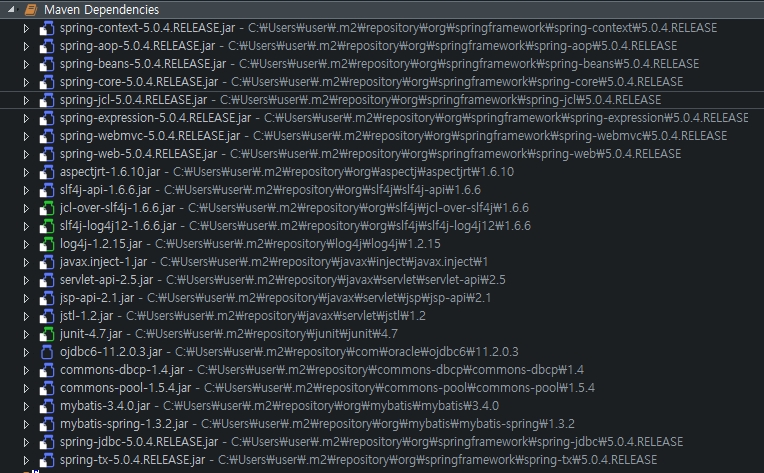
3) pom.xml

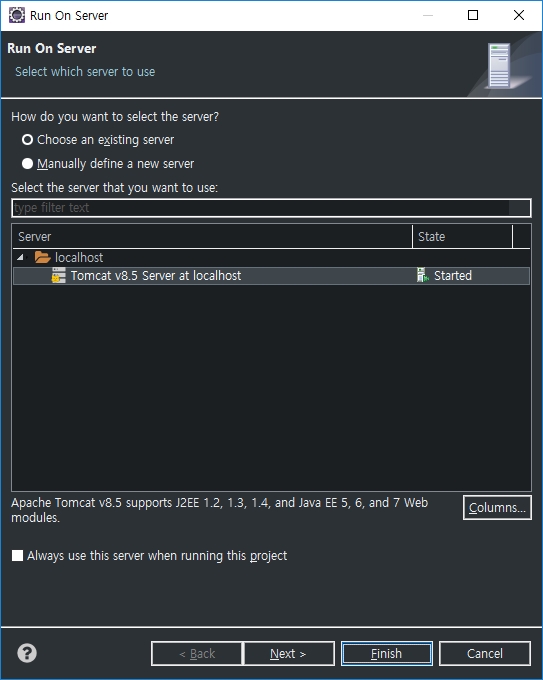
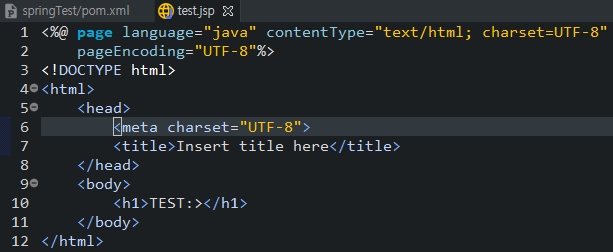


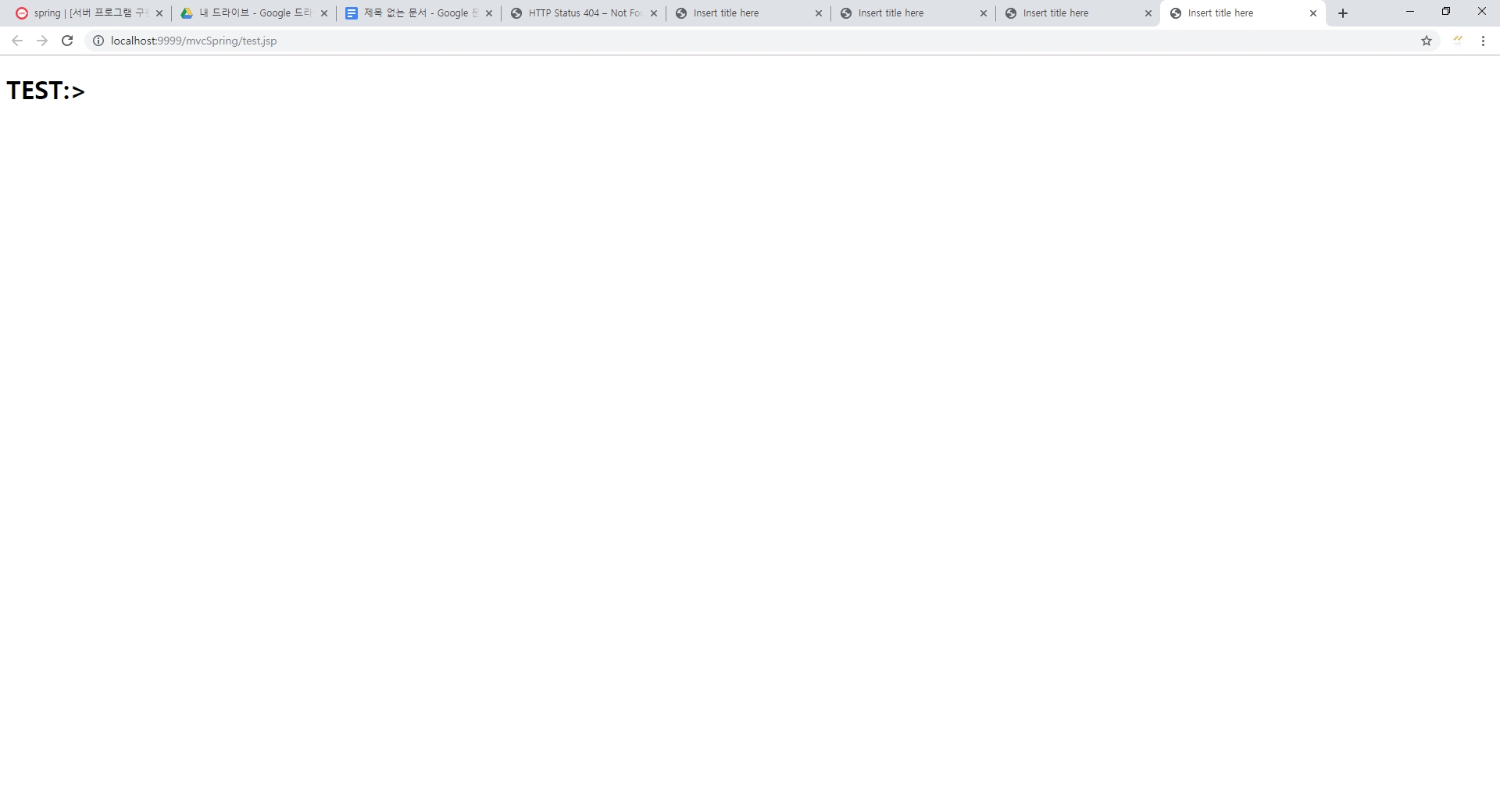
4) mysql 커넥터 설정



5) mysql의 maven위치 확인



6) 톰킷 웹서버 실행 여부 확인



spring project라는 전제 하에, webapp에 jsp파일을 만들고, 표현하고자 하는 내용 작성

-> run을 하게되면 server를 선택할 수 있고, 알맞은 서버 선택 후 finish 클릭

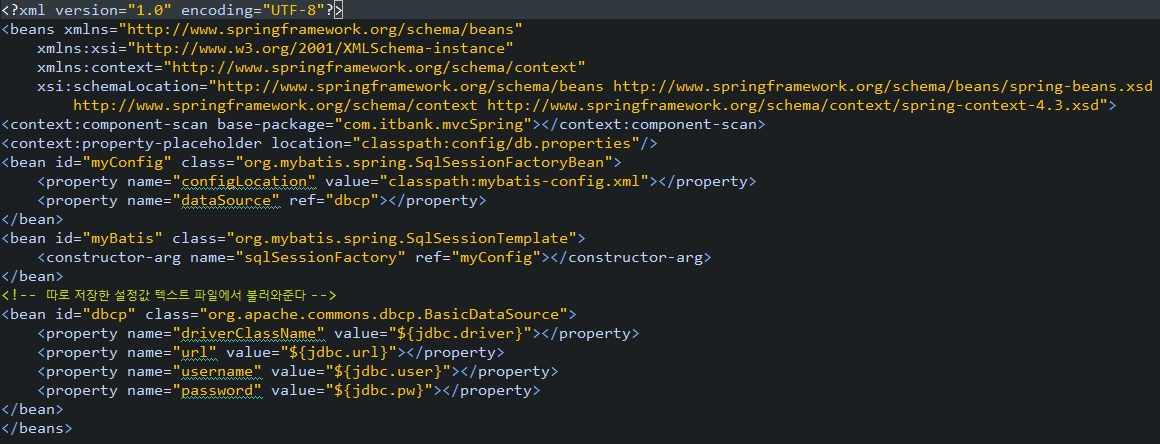
( 띄우고자 하는 브라우저는 eclipse 메뉴에서 선택 가능!)

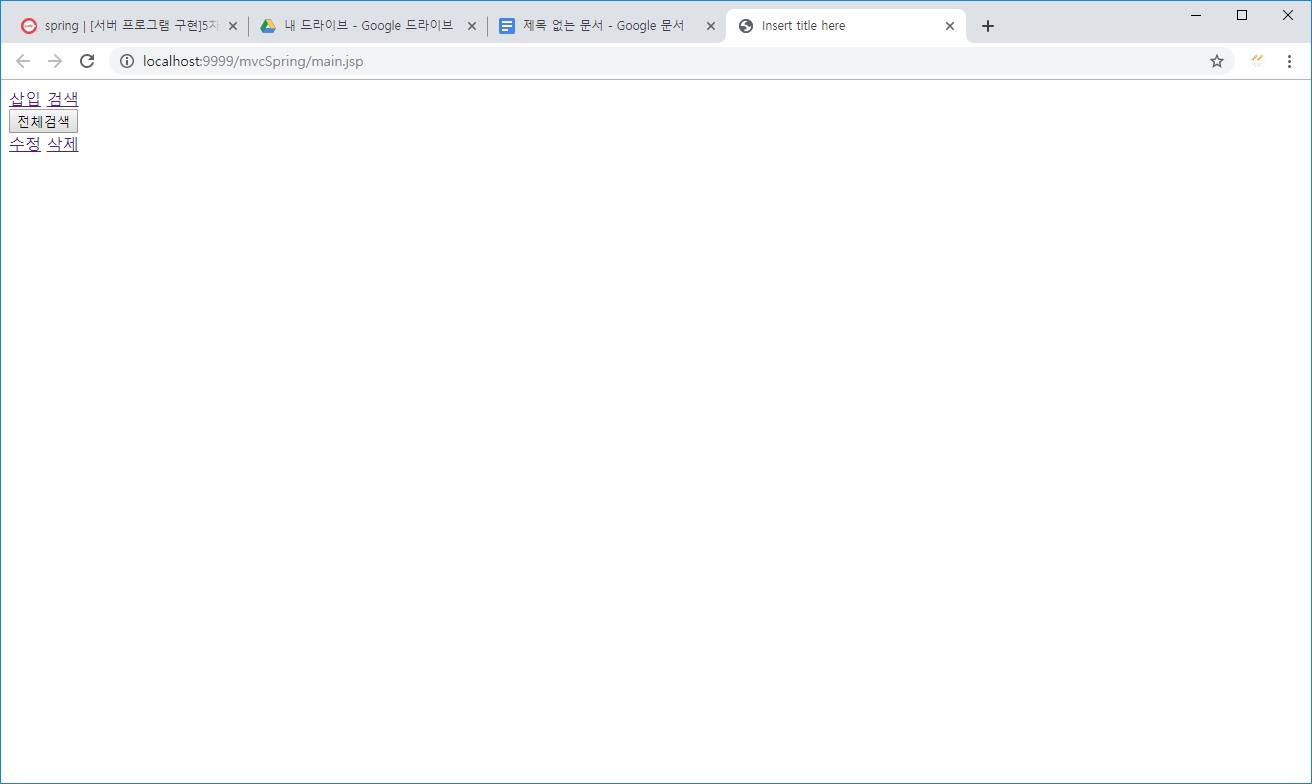
2. 공통 모듈 구현하기

1. dao, dto가 들어가는 위치

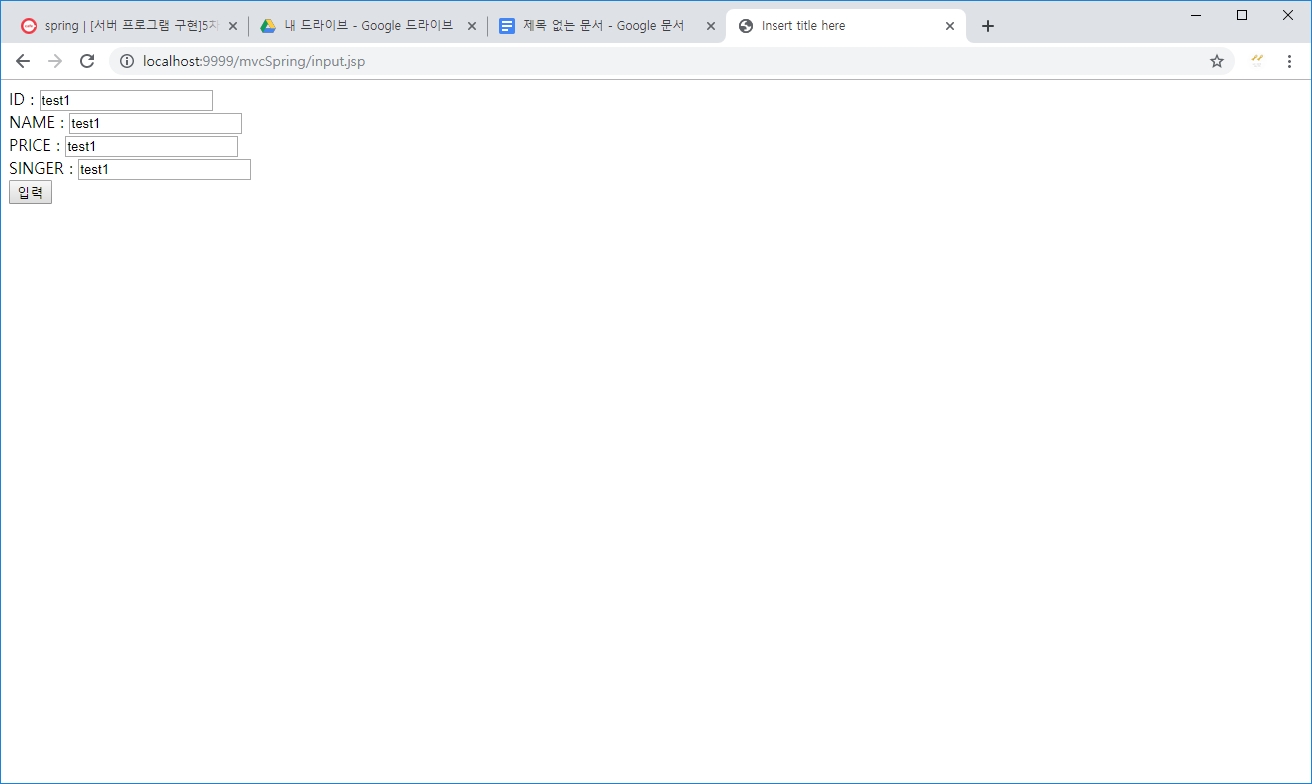
 src/main/java 아래에 자동 생성되는 패키지 안!

2) context.xml (music.xml)



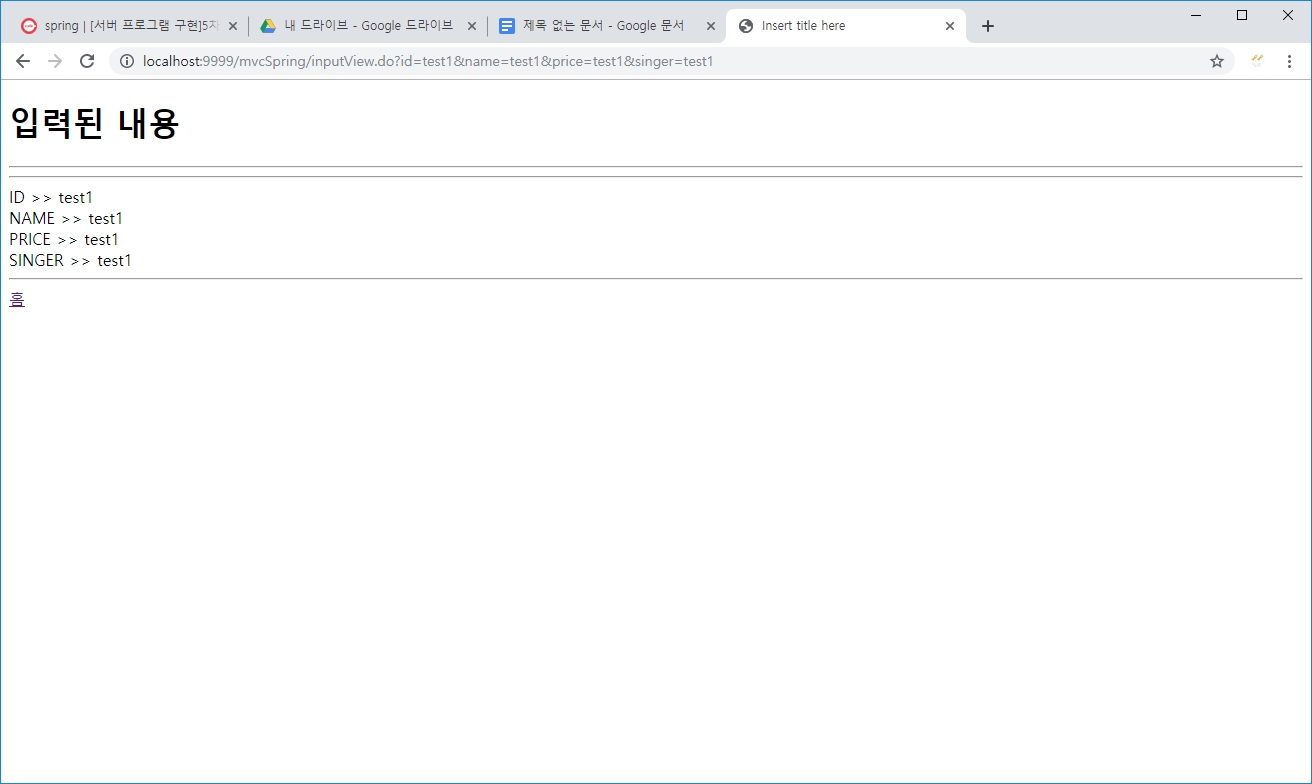
3. 서버 프로그램 구현하기/ 배치 프로그램 구현하기

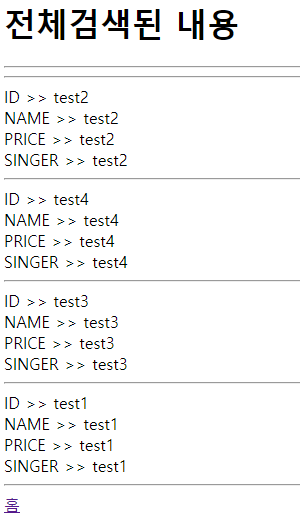
처음 메인 화면

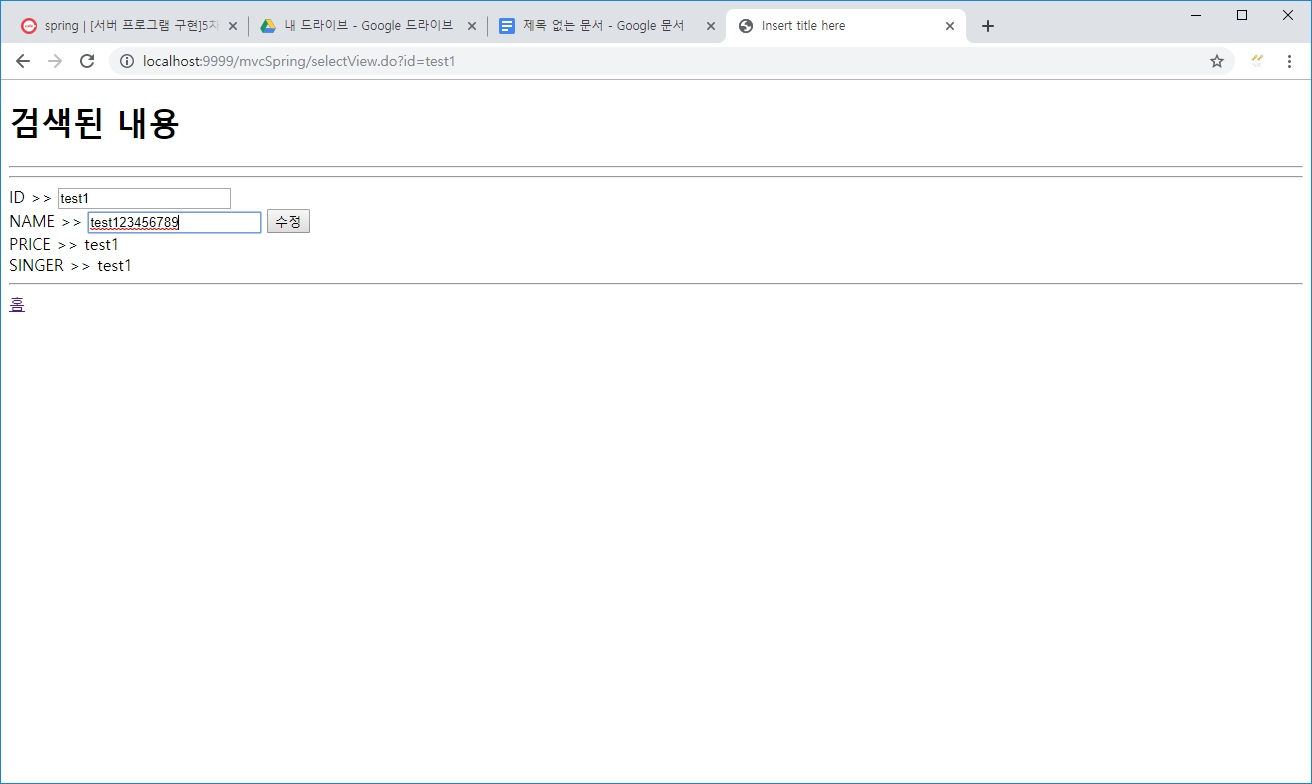
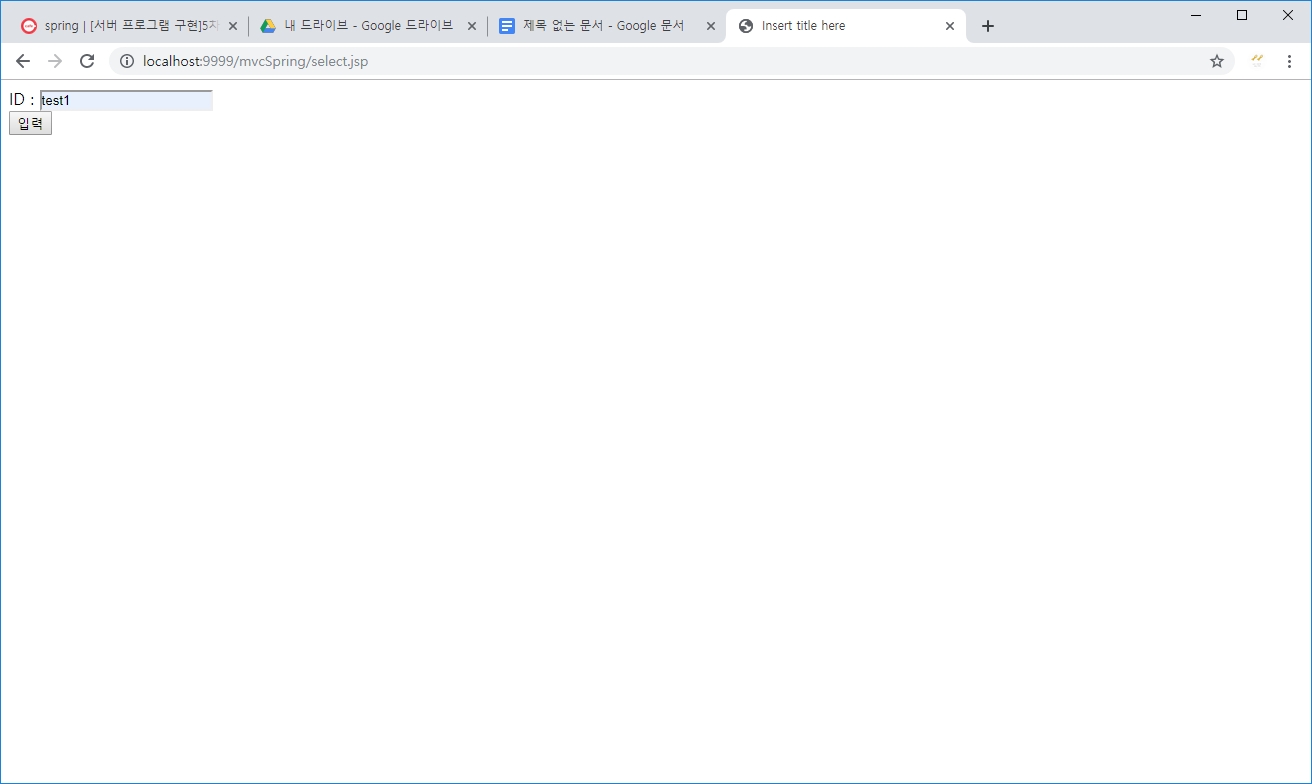
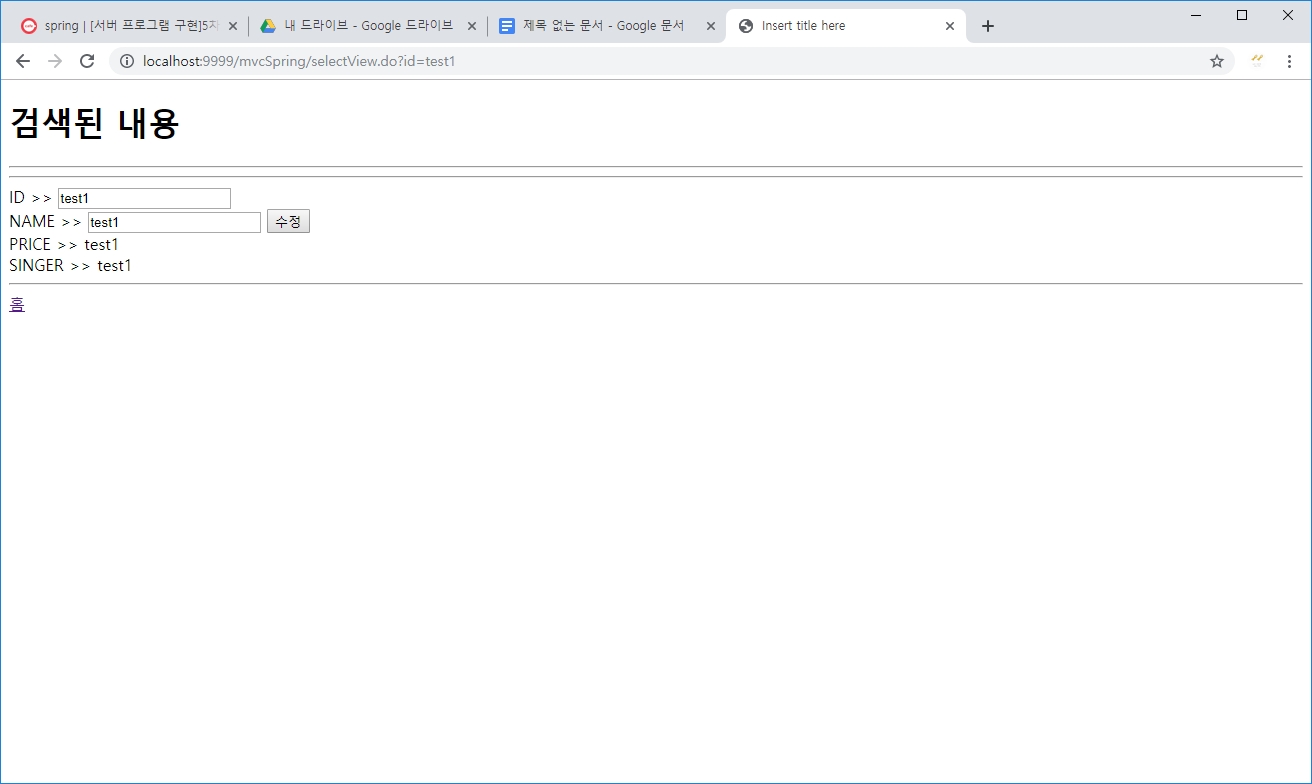
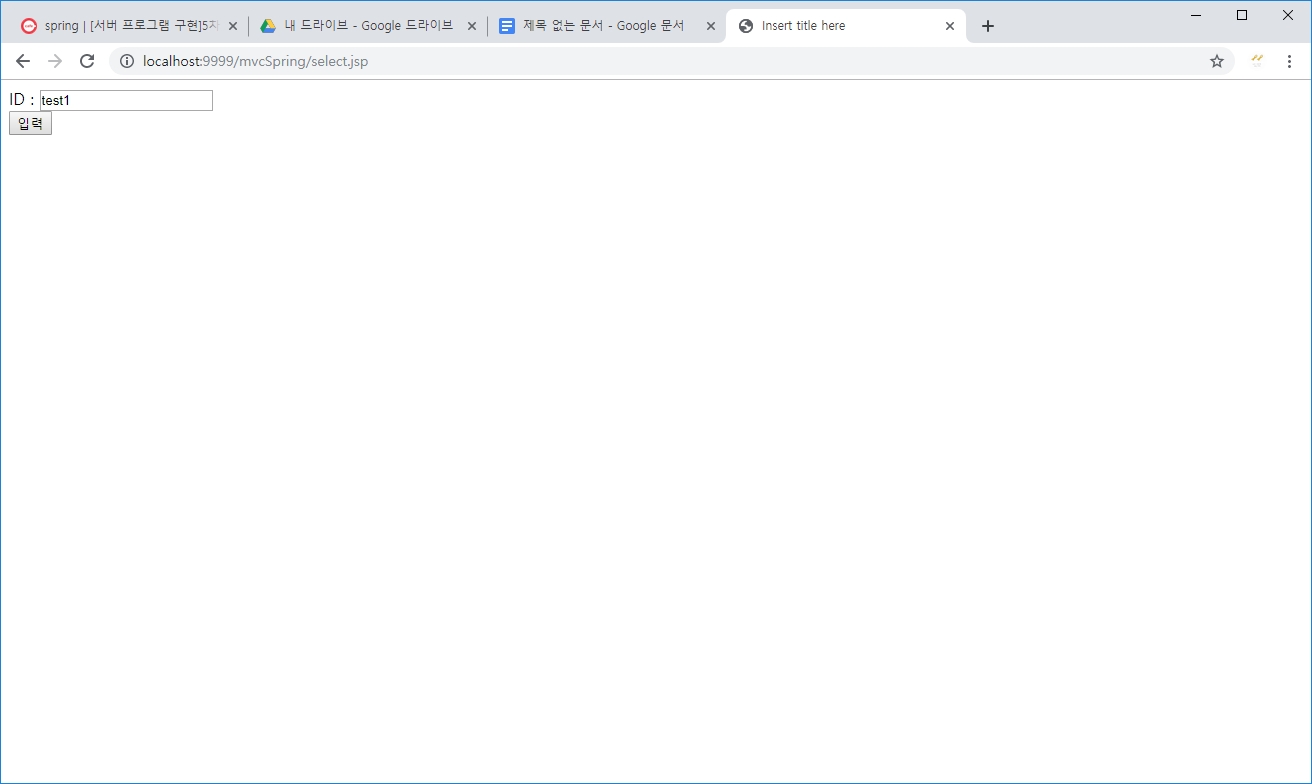


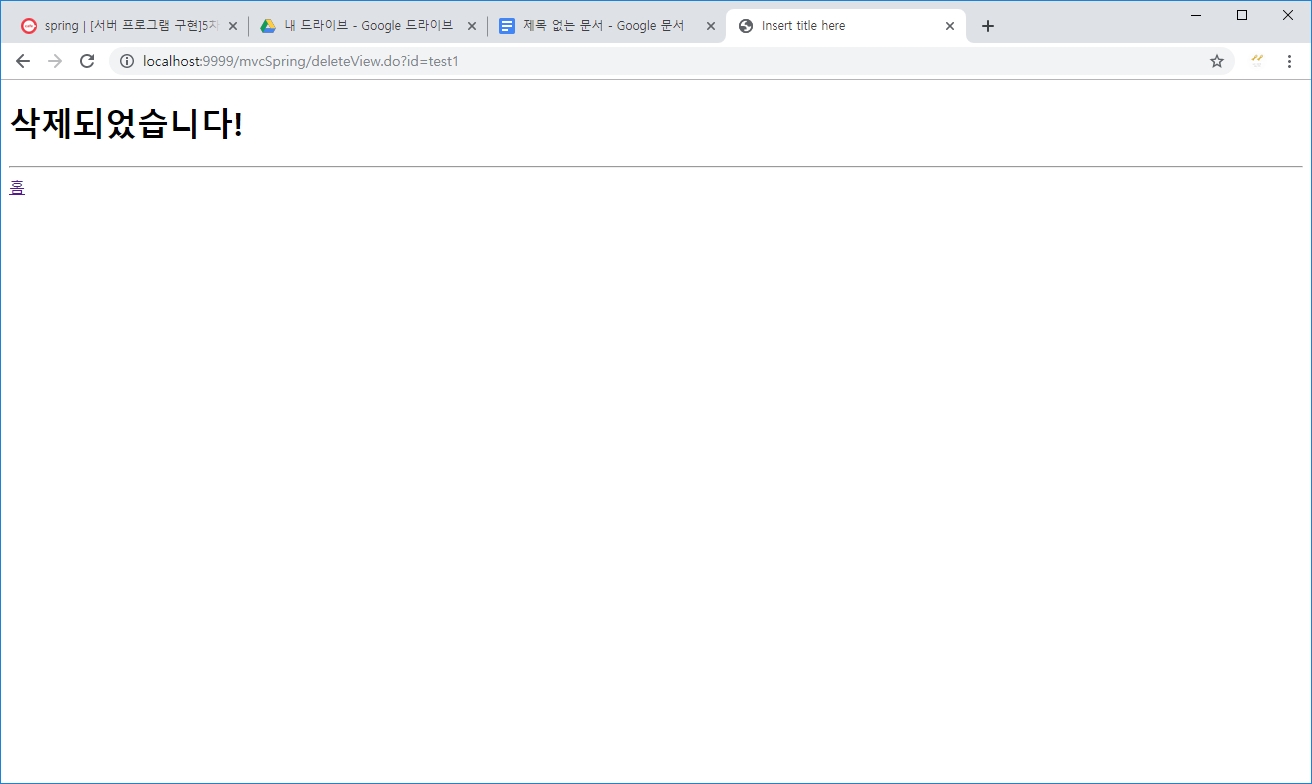
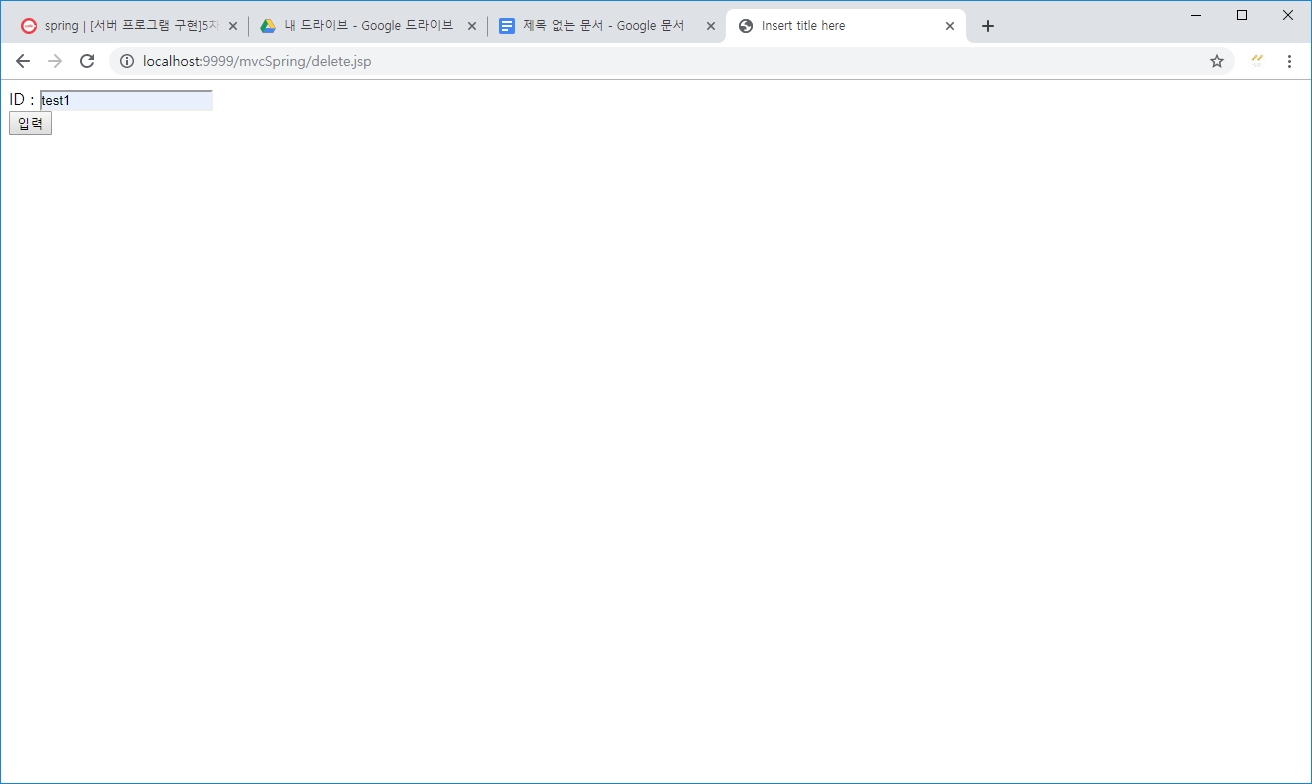
클라이언트의 요청을 받는 부분!

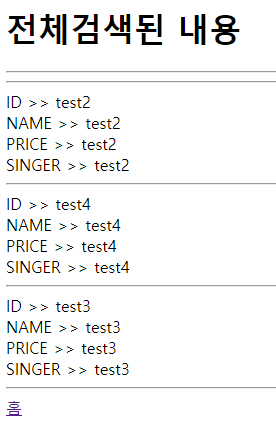
id와 name, price, singer를 받아서 controller로 넘겨준다

controller는 mapping된 views를 보여줌!! 결과가 보여지는 부분

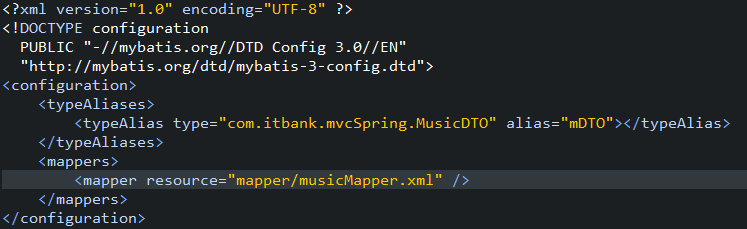


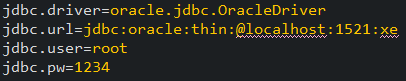






* mybatis 설정 파일 캡처



 >> db.properties

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 소스코드 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MusicController

package com.itbank.mvcSpring;

import java.util.List;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Controller;

import org.springframework.ui.Model;

import org.springframework.web.bind.annotation.RequestMapping;

// annotation 으로 컨트롤러 설정

@Controller

public class MusicController {

@Autowired

MusicDAO dao;

// mapping 해주는 위치는 views아래의 jsp파일명과 동일하며, .뒤에 있는 값은 보여지지 않는다!

// 다른 메서드로 같은 jsp파일을 mapping할 때의 위치값 개별 부여 위함

@RequestMapping("inputView.do")

public void insert(MusicDTO musicDTO) {

dao.insert(musicDTO);

}

@RequestMapping("updateView.do")

public void update(MusicDTO musicDTO) {

dao.update(musicDTO);

}

@RequestMapping("deleteView.do")

public void delete(MusicDTO musicDTO) {

dao.delete(musicDTO);

}

@RequestMapping("selectView.do")

public void select(MusicDTO musicDTO, Model model) {

List<MusicDTO> list = dao.select(musicDTO);

model.addAttribute("list", list);

}

@RequestMapping("selectAllView.do")

public void selectAll(Model model) {

List<MusicDTO> list = dao.selectAll();

model.addAttribute("list2", list); // list2로 list에 담긴 내용을 저장함 따라서, views에서 보여줄 때, list2로 foreach 돌리기!

}

}

MusicDAO

package com.itbank.mvcSpring;

import java.util.List;

import org.mybatis.spring.SqlSessionTemplate;

import org.springframework.beans.factory.annotation.Autowired;

import org.springframework.stereotype.Repository;

//@Component 부모

//@Service 자식 보통의 bean에서 사용

@Repository // 아무데서나 싱글톤으로 만든 dao를 쓸 수 있다! 보통 db용 bean으로 사용

public class MusicDAO {

// 설정값에서 myBatis의 주소를 불러와서 sqlsessiontemplate라는 타입으로 찾아서 autowired가 설정해줌

@Autowired

SqlSessionTemplate my;

public void insert(MusicDTO musicDTO) {

my.insert("mDAO.insert", musicDTO);

}

public List<MusicDTO> select(MusicDTO musicDTO) {

return my.selectList("mDAO.select", musicDTO);

}

// arraylist따로 만들지 않아도 알아서 list만들어서 넘겨줌

public List<MusicDTO> selectAll() {

return my.selectList("mDAO.selectAll");

}

public void update(MusicDTO musicDTO) {

my.update("mDAO.update", musicDTO);

}

public void delete(MusicDTO musicDTO) {

my.delete("mDAO.delete", musicDTO);

}

}

MusicDTO

package com.itbank.mvcSpring;

public class MusicDTO {

String id;

String name;

String price;

String singer;

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getPrice() {

return price;

}

public void setPrice(String price) {

this.price = price;

}

public String getSinger() {

return singer;

}

public void setSinger(String singer) {

this.singer = singer;

}

}

db.properties

jdbc.driver=oracle.jdbc.OracleDriver

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

jdbc.user=root

jdbc.pw=1234

musicMapper

<?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="mDAO">

<select id="select" parameterType="mDTO" resultMap="rsMap">

SELECT \* FROM music WHERE id = #{id}

</select>

<resultMap type="mDTO" id="rsMap">

<id property="id" column="id" />

<result property="name" column="name" />

<result property="price" column="price" />

<result property="singer" column="singer" />

</resultMap>

<select id="selectAll" resultMap="rsMap">

SELECT \* FROM music

</select>

<insert id="insert" parameterType="mDTO">

INSERT INTO music VALUES(#{id},#{name},#{price},#{singer})

</insert>

<update id="update" parameterType="mDTO">

UPDATE music SET name = #{name} WHERE id = #{id}

</update>

<delete id="delete" parameterType="mDTO">

DELETE FROM music WHERE

id = #{id}

</delete>

</mapper>

music.xml

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

<beans xmlns="http://www.springframework.org/schema/beans"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xmlns:context="http://www.springframework.org/schema/context"

xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema/beans/spring-beans.xsd

http://www.springframework.org/schema/context http://www.springframework.org/schema/context/spring-context-4.3.xsd">

<context:component-scan base-package="com.itbank.mvcSpring"></context:component-scan>

<context:property-placeholder location="classpath:config/db.properties"/>

<bean id="myConfig" class="org.mybatis.spring.SqlSessionFactoryBean">

<property name="configLocation" value="classpath:mybatis-config.xml"></property>

<property name="dataSource" ref="dbcp"></property>

</bean>

<bean id="myBatis" class="org.mybatis.spring.SqlSessionTemplate">

<constructor-arg name="sqlSessionFactory" ref="myConfig"></constructor-arg>

</bean>

<!-- 따로 저장한 설정값 텍스트 파일에서 불러와준다 -->

<bean id="dbcp" class="org.apache.commons.dbcp.BasicDataSource">

<property name="driverClassName" value="${jdbc.driver}"></property>

<property name="url" value="${jdbc.url}"></property>

<property name="username" value="${jdbc.user}"></property>

<property name="password" value="${jdbc.pw}"></property>

</bean>

</beans>

mybatis-config

<?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>

<typeAliases>

<typeAlias type="com.itbank.mvcSpring.MusicDTO" alias="mDTO"></typeAlias>

</typeAliases>

<mappers>

<mapper resource="mapper/musicMapper.xml" />

</mappers>

</configuration>

deleteView

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

<h1>삭제되었습니다!</h1>

<hr>

<a href="main.jsp">홈</a>

</body>

</html>

inpurView

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

<h1>입력된 내용</h1><hr><hr>

ID >> ${musicDTO.id}<br>

NAME >> ${musicDTO.name}<br>

PRICE >> ${musicDTO.price}<br>

SINGER >> ${musicDTO.singer}<br>

<hr>

<a href="main.jsp">홈</a>

</body>

</html>

selectAllView

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<h1>전체검색된 내용</h1><hr><hr>

<c:forEach var="dto" items="${list2}">

ID >> ${dto.id}<br>

NAME >> ${dto.name}<br>

PRICE >> ${dto.price}<br>

SINGER >> ${dto.singer}<br>

<hr>

</c:forEach>

<a href="main.jsp">홈</a>

selectView

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

<h1>검색된 내용</h1><hr><hr>

<c:forEach var="dto" items="${list}">

<form action="updateView.do">

ID >> <input type="text" value="${dto.id}" name="id"><br>

NAME >> <input type="text" value="${dto.name}" name="name"> <input type="submit" value="수정"><br>

</form>

PRICE >> ${dto.price}<br>

SINGER >> ${dto.singer}<br>

</c:forEach>

<hr>

<a href="main.jsp">홈</a>

</body>

</html>

updateView

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

<h1>수정된 내용</h1><hr><hr>

ID >> ${musicDTO.id}<br>

NAME >> ${musicDTO.name}<br>

PRICE >> ${musicDTO.price}<br>

SINGER >> ${musicDTO.singer}<br>

<hr>

<a href="main.jsp">홈</a>

</body>

</html>

web.xml

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

<web-app version="2.5" xmlns="http://java.sun.com/xml/ns/javaee"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://java.sun.com/xml/ns/javaee https://java.sun.com/xml/ns/javaee/web-app\_2\_5.xsd">

<!-- The definition of the Root Spring Container shared by all Servlets and Filters -->

<context-param>

<param-name>contextConfigLocation</param-name>

<param-value>classpath:music.xml</param-value>

</context-param>

<!-- Creates the Spring Container shared by all Servlets and Filters -->

<listener>

<listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>

</listener>

<!-- Processes application requests -->

<servlet>

<servlet-name>appServlet</servlet-name>

<servlet-class>org.springframework.web.servlet.DispatcherServlet</servlet-class>

<init-param>

<param-name>contextConfigLocation</param-name>

<param-value>/WEB-INF/spring/appServlet/servlet-context.xml</param-value>

</init-param>

<load-on-startup>1</load-on-startup>

</servlet>

<servlet-mapping>

<servlet-name>appServlet</servlet-name>

<url-pattern>/</url-pattern>

</servlet-mapping>

</web-app>

delete

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

<form action="deleteView.do">

ID : <input type="text" name="id"><br>

<input type="submit" value="입력">

</form>

</body>

</html>

input

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

<form action="inputView.do">

ID : <input type="text" name="id"><br>

NAME : <input type="text" name="name"><br>

PRICE : <input type="text" name="price"><br>

SINGER : <input type="text" name="singer"><br>

<input type="submit" value="입력">

</form>

</body>

</html>

main

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

<a href="input.jsp">삽입</a>

<a href="select.jsp">검색</a>

<form action="selectAllView.do"><input type="submit" value="전체검색"></form>

<a href="select.jsp">수정</a>

<a href="delete.jsp">삭제</a>

</body>

</html>

select

<%@ page language="java" contentType="text/html; charset=UTF-8"

pageEncoding="UTF-8"%>

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title>Insert title here</title>

</head>

<body>

<form action="selectView.do">

ID : <input type="text" name="id"><br>

<input type="submit" value="입력">

</form>

</body>

</html>

pom.xml

<?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 https://maven.apache.org/maven-v4\_0\_0.xsd">

<modelVersion>4.0.0</modelVersion>

<groupId>com.itbank</groupId>

<artifactId>mvcSpring</artifactId>

<name>springTest</name>

<packaging>war</packaging>

<version>1.0.0-BUILD-SNAPSHOT</version>

<properties>

<java-version>1.8</java-version>

<org.springframework-version>5.0.4.RELEASE</org.springframework-version>

<org.aspectj-version>1.6.10</org.aspectj-version>

<org.slf4j-version>1.6.6</org.slf4j-version>

</properties>

<repositories>

<repository>

<id>mvn2</id>

<url>http://repo1.maven.org/maven2/</url>

<releases>

<enabled>true</enabled>

</releases>

<snapshots>

<enabled>true</enabled>

</snapshots>

</repository>

<repository>

<id>oracle</id>

<name>ORACLE JDBC Repository</name>

<url>http://maven.jahia.org/maven2</url>

</repository>

<repository>

<id>codelds</id>

<url>http://code.lds.org/nexus/content/groups/main-repo</url>

</repository>

</repositories>

<dependencies>

<!-- Spring -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-context</artifactId>

<version>${org.springframework-version}</version>

<exclusions>

<!-- Exclude Commons Logging in favor of SLF4j -->

<exclusion>

<groupId>commons-logging</groupId>

<artifactId>commons-logging</artifactId>

</exclusion>

</exclusions>

</dependency>

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>${org.springframework-version}</version>

</dependency>

<!-- AspectJ -->

<dependency>

<groupId>org.aspectj</groupId>

<artifactId>aspectjrt</artifactId>

<version>${org.aspectj-version}</version>

</dependency>

<!-- Logging -->

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-api</artifactId>

<version>${org.slf4j-version}</version>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>jcl-over-slf4j</artifactId>

<version>${org.slf4j-version}</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>org.slf4j</groupId>

<artifactId>slf4j-log4j12</artifactId>

<version>${org.slf4j-version}</version>

<scope>runtime</scope>

</dependency>

<dependency>

<groupId>log4j</groupId>

<artifactId>log4j</artifactId>

<version>1.2.15</version>

<exclusions>

<exclusion>

<groupId>javax.mail</groupId>

<artifactId>mail</artifactId>

</exclusion>

<exclusion>

<groupId>javax.jms</groupId>

<artifactId>jms</artifactId>

</exclusion>

<exclusion>

<groupId>com.sun.jdmk</groupId>

<artifactId>jmxtools</artifactId>

</exclusion>

<exclusion>

<groupId>com.sun.jmx</groupId>

<artifactId>jmxri</artifactId>

</exclusion>

</exclusions>

<scope>runtime</scope>

</dependency>

<!-- @Inject -->

<dependency>

<groupId>javax.inject</groupId>

<artifactId>javax.inject</artifactId>

<version>1</version>

</dependency>

<!-- Servlet -->

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>servlet-api</artifactId>

<version>2.5</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>javax.servlet.jsp</groupId>

<artifactId>jsp-api</artifactId>

<version>2.1</version>

<scope>provided</scope>

</dependency>

<dependency>

<groupId>javax.servlet</groupId>

<artifactId>jstl</artifactId>

<version>1.2</version>

</dependency>

<!-- Test -->

<dependency>

<groupId>junit</groupId>

<artifactId>junit</artifactId>

<version>4.7</version>

<scope>test</scope>

</dependency>

<dependency>

<groupId>com.oracle</groupId>

<artifactId>ojdbc6</artifactId>

<version>11.2.0.3</version>

</dependency>

<dependency>

<groupId>commons-dbcp</groupId>

<artifactId>commons-dbcp</artifactId>

<version>1.4</version>

</dependency>

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis</artifactId>

<version>3.4.0</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.mybatis/mybatis-spring -->

<dependency>

<groupId>org.mybatis</groupId>

<artifactId>mybatis-spring</artifactId>

<version>1.3.2</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.spring framework/spring-jdbc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>5.0.4.RELEASE</version>

</dependency>

</dependencies>

<build>

<plugins>

<plugin>

<artifactId>maven-eclipse-plugin</artifactId>

<version>2.9</version>

<configuration>

<additionalProjectnatures>

<projectnature>org.springframework.ide.eclipse.core.springnature</projectnature>

</additionalProjectnatures>

<additionalBuildcommands>

<buildcommand>org.springframework.ide.eclipse.core.springbuilder</buildcommand>

</additionalBuildcommands>

<downloadSources>true</downloadSources>

<downloadJavadocs>true</downloadJavadocs>

</configuration>

</plugin>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-compiler-plugin</artifactId>

<version>2.5.1</version>

<configuration>

<source>1.6</source>

<target>1.6</target>

<compilerArgument>-Xlint:all</compilerArgument>

<showWarnings>true</showWarnings>

<showDeprecation>true</showDeprecation>

</configuration>

</plugin>

<plugin>

<groupId>org.codehaus.mojo</groupId>

<artifactId>exec-maven-plugin</artifactId>

<version>1.2.1</version>

<configuration>

<mainClass>org.test.int1.Main</mainClass>

</configuration>

</plugin>

</plugins>

</build>

</project>