

# 응용 프로그래밍 및 화면 구현

2023/02/20 라성준

# 문제 1. 프로젝트 생성



## Project

☒ Gradle - Groovy ☐ Gradle - Kotlin ☒ Java ☐ Kotlin ☐ Groovy  
☐ Maven

## Spring Boot

☐ 3.0.3 (SNAPSHOT) ☐ 3.0.2 ☐ 2.7.9 (SNAPSHOT) ☒ 2.7.8

## Project Metadata

Group   
Artifact   
Name   
Description   
Package name   
Packaging ☒ Jar ☐ War  
Java ☐ 19 ☐ 17 ☒ 11 ☐ 8

## Language

## Dependencies

ADD DEPENDENCIES... CTRL + B

### Spring Boot DevTools DEVELOPER TOOLS

Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

### Lombok DEVELOPER TOOLS

Java annotation library which helps to reduce boilerplate code.

### Spring Web WEB

Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

### Thymeleaf TEMPLATE ENGINES

A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.

### MySQL Driver SQL

MySQL JDBC driver.

## 문제2. 화면 구현

Todo

Ready

Doing

Done

#101  
내용1  
20xx-01-01

#102  
내용2  
20xx-01-01

추가

### ▶ 아이템 드래킹 처리

```
$(function(){
  $('article').sortable({
    connectWith: "article",
    scroll: false,
    helper: "clone",
    receive: function(e, ui){
      let todoNum = $(ui.item).attr('data-no');
      let todoStatus = $(this).attr('data-status');

      const content = {"todoNum":todoNum,"todoStatus":todoStatus}
```

## 2.화면 구현

Todo

Ready

#0  
2023/02/20 수행평가  
2023년 2월 20일 X

Doing

#0  
2023/02/20 수행평가22  
2023년 2월 20일 X

Done

2023/02/20 수행평가22

추가

```
$('#btnAdd').click(function(){
    let value = $('input[name=todo]').val();
    const date = new Date();
    const dateS = date.toLocaleDateString('ko', {
        day: 'numeric',
        month: 'short',
        year: 'numeric'
    });

    jsonData = {"todoContent":value};

    const xhr = new XMLHttpRequest();
    xhr.open("POST", "/Todo/todoInsert", true);
    xhr.responseType = "json";

    xhr.onreadystatechange = function() {
        if(xhr.readyState == XMLHttpRequest.DONE) {
            if(xhr.status != 200) alert("Request fail...")
            else {
                const data = xhr.response;
            }
        }
    }
});
```

- ▶ 하단 리스트 아이템 추가 기능
- ▶ 리스트별 아이템 삭제 기능

## 문제3. 테이블 설계

#	이름	데이터 유형	길이/설정	부호 ...	NULL ...	0으로 ...	기본값
1	todoNUM	SMALLINT	5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
2	todoContent	TEXT		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
3	todoRdate	DATE		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL
4	todoStatus	TINYINT	3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	NULL

## 문제4. 기능 구현

### Todo

#### Ready

- #1  
수행평가  
2023-02-20 X
- #4  
□ □ □ □ □  
2023-02-20 X
- #5  
□ □ □ □ □  
2023-02-20 X

#### Doing

- #2  
수행평가2  
2023-02-20 X

#### Done

- #3  
□  
2023-02-20 X

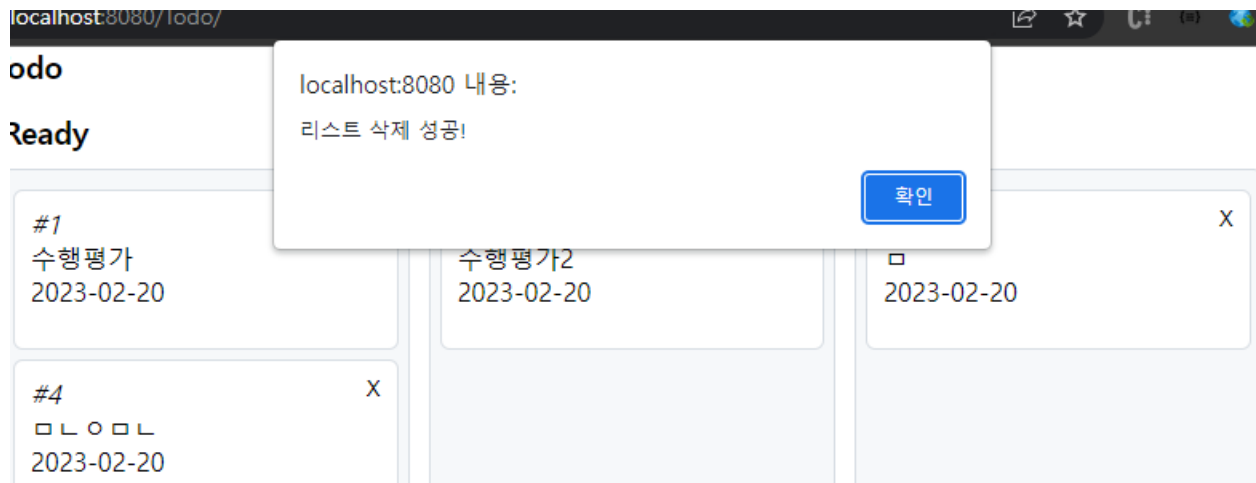
todoNUM	todoContent	todoRdate	todoStatus
1	수행평가	2023-02-20	1
2	수행평가2	2023-02-20	2
3	□	2023-02-20	3
4	□ □ □ □ □	2023-02-20	1
5	□ □ □ □ □	2023-02-20	1

```
$(function(){
    $('article').sortable({
        connectWith: "article",
        scroll: false,
        helper: "clone",
        receive: function(e, ui){
            let todoNum = $(ui.item).attr('data-no');
            let todoStatus = $(this).attr('data-status');

            const content = {"todoNum":todoNum,"todoStatus":todoStatus}

            $.ajax({
                url:"/Todo/todoUpdate",
                type:"POST",
                contentType: 'application/json',
                data:JSON.stringify(content),
                dataType:"json",
                success: (data)=>{
                    if(data.result > 0) {
                        alert('상태 변화 성공!');
                    } else {
                        alert('상태 변화 실패!');
                    }
                }
            })
        }
    })
})
```

## 문제4. 기능 구현



```
$.ajax({
  url: "/Todo/todoDelete",
  type: "POST",
  contentType: 'application/json',
  data: JSON.stringify(content),
  dataType: "json",
  success: (data) => {
    if (data.result > 0) {
      alert('리스트 삭제 성공!');
      $(this).parent().remove();
    } else {
      alert('리스트 삭제 실패!');
    }
  }
},
```

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE mapper
  PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"
  "https://mybatis.org/dtd/mybatis-3-mapper.dtd">
<mapper namespace="kr.co.todo.dao.MainDAO">

  <insert id="insertTodoList" parameterType="kr.co.todo.vo.TODOVO"
    useGeneratedKeys="true" keyProperty="todoNum" keyColumn="todoNum">
    INSERT INTO todo SET todoContent = #{todoContent} , todoRdate = NOW();
  </insert>

  <!-- todo 리스트 불러오기 -->
  <select id="selectTodoList" resultType="kr.co.todo.vo.TODOVO">
    SELECT * FROM todo;
  </select>

  <!-- todo 리스트 업데이트 쿼리문 -->
  <update id="updateTodoList">
    UPDATE todo SET todoStatus = ${todoStatus} WHERE todoNum = ${todoNum};
  </update>

  <!-- delete -->
  <delete id="deleteTodoList" parameterType="kr.co.todo.vo.TODOVO">
    DELETE FROM todo WHERE todoNum = #{todoNum};
  </delete>
</mapper>
```