



# 응용 프로그래밍 및 화면 구현

통합 구현

---

심규영

# 목차

- 1) 프로젝트 생성
- 2) 화면 구현
- 3) 테이블 설계
- 4) 기능 구현

# 1) 프로젝트 생성

## Project

☒ Gradle - Groovy  
☐ Gradle - Kotlin ☐ Maven

## Language

☒ Java ☐ Kotlin ☐ Groovy

## 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

## Dependencies

ADD DEPENDENCIES... CTRL + B

### Lombok DEVELOPER TOOLS

Java annotation library which helps to reduce boilerplate code.

### Spring Boot DevTools DEVELOPER TOOLS

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

### 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.

### MyBatis Framework SQL

Persistence framework with support for custom SQL, stored procedures and advanced mappings. MyBatis couples objects with stored procedures or SQL statements using a XML descriptor or annotations.

### MySQL Driver SQL

MySQL JDBC driver.

## 2) 화면 구현

화면 레이아웃 작업

Todo

Ready

Doing

Done

#101  
내용1  
20xx-01-01

X

#102  
내용2  
20xx-01-01

X

추가

## 2) 화면 구현

아이템 드래킹 처리(jQueryUI sortable API 활용)

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

            console.log("no : "+no);
            console.log("value : "+value);
        }
    });
});
```

Todo

Ready

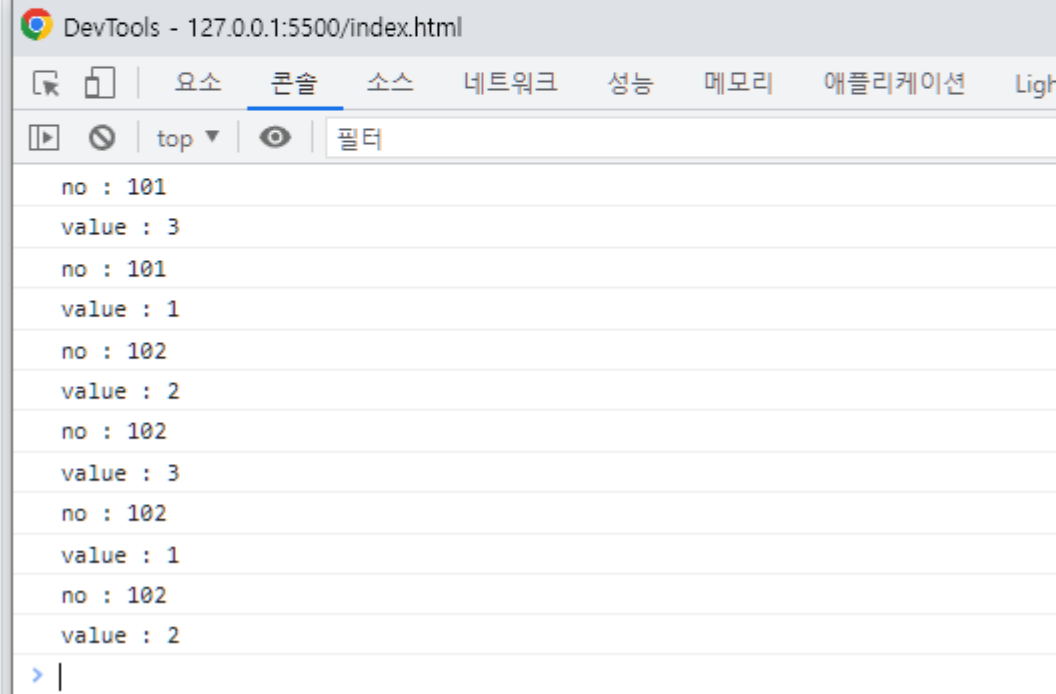
Doing

#101  
내용1  
20xx-01-01

X

#102  
내용2  
20xx-01-01

X



## 2) 화면 구현

하단 리스트 아이템 추가 기능  
리스트별 아이템 삭제 기능

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

  let item = `<div class='item'>
    <button class="del">X</button>
    <em class="tit">#200</em>
    <p>`+value+`</p>
    <span class="date">`+dateS+`</span>
  </div>`;

  $('#.ready').append(item);
});

$(document).on('click', '.del', function(){
  $(this).parent().remove();
});
```

### Todo

#### Ready

#200 X  
내용 3  
2023년 2월 20일

#200 X  
내용 4  
2023년 2월 20일

#### Doing

#102 X  
내용2  
20xx-01-01

#### Done

#101 X  
내용1  
20xx-01-01

### 3) 테이블 설계

기본

옵션

인덱스 (1)

외래 키 (0)

제약 조건 확인 (0)

분할

CREATE 코드

ALTER 코드

이름:

todo

코멘트:


열:

+ 추가

✕ 제거

▲ 위로

▼ 아래로

#	이름	데이터 유형	길이/설정	부호 ...	NULL 허용	0으로 채...	기본값	코멘트
 1	todoNum	SMALLINT	5	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	AUTO_INCREMENT	todo 번호
2	todoContent	TEXT		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음	Todo 내용
3	todoRdate	DATE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	기본값 없음	작성 날짜
4	todoStatus	TINYINT	3	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'1'	1 => ready2 => doing3 => done

## 4) 기능 구현

메인화면 아이템 Select 처리

```
<!-- 2023/02/20 // 심규영 // todo 리스트 불러오기 -->
<select id="selectTodoList" resultType="kr.co.todo.vo.TODOVO">
    SELECT * FROM todo;
</select>
```



```
/**
 * 2023/02/20 // 심규영 // 투두 리스트 불러오기
 * 리스트 불러온후 상태 값에 따라 분리
 * @return 1, 2, 3
 */
1 usage
public Map<Integer, List<TODOVO>> selectTodoList() {
    List<TODOVO> list = dao.selectTodoList();
    return list.stream().collect(Collectors.groupingBy(TODOVO::getTodoStatus));
}
```



```
<h3>Doing</h3>
<article class="doing" data-status="2">
    <th:block th:each="article:${data[2]}">
        <div class="item" th:data-no="${article.todoNum}">
            <button class="del">X</button>
            <em class="tit" th:text="${'#'+ article.todoNum}">#100</em>
            <p th:text="${article.todoContent}">내용1</p>
            <span class="date" th:text="${article.todoRdate}">20xx-01-01</span>
        </div>
    </th:block>
</article>
```

Todo

Ready

Doing

Done

The screenshot shows a web application with three columns: Ready, Doing, and Done. Each column contains a todo item with a delete button (X). The 'Doing' column is highlighted. Below the web application, a HeidiSQL window is open, showing the 'todo' table in the 'java2db' database. The table contains three rows of data.

todoNum	todoContent	todoRdate	todoStatus
1	할일 1	2023-02-20	1
2	할일 2	2023-02-20	2
3	할일 3	2023-02-20	3



## 4) 기능 구현

아이템 이동시 update 처리

```
$( '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!');
        }
      }
    });
  }
});
```

Todo

Ready

상태 변화 성공!

확인

#1  
할일 1  
2023-02-20

#3  
할일 3  
2023-02-20

#2  
할일 2  
2023-02-20

va2db#todo# - HeidiSQL 12.1.0.6537

검색 쿼리 도구 이동 도움말

이스 필터 테이블 필터

호스트: 127.0.0.1 데이터베이스: java2db 테이블: todo 데이터 쿼리+

child 48.0 KiB  
department 16.0 KiB  
member 32.0 KiB  
membervo 16.0 KiB  
parent 32.0 KiB

java2db.todo: 3 행 (총) (대략적)

todoNum	todoContent	todoRdate	todoStatus
1	할일 1	2023-02-20	1
2	할일 2	2023-02-20	3
3	할일 3	2023-02-20	3

## 4) 기능 구현

아이템 추가시 insert 처리

```
$('#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;

                if(data.result > 0) {
                    // 성공
                    alert('게시물 생성에 성공 하셨습니다.');
```



```
                    let item = `<div class='item'>
                        <button class="del">X</button>
                        <em class="tit">#'+data.vo.todoNum+'</em>
                        <p>'+value+'</p>
                        <span class="date">'+dateS+'</span>
                    </div>`;

                    $('#.ready').append(item);
                } else {
                    // 실패
                    alert('게시물 생성에 실패 하였습니다.');
```

```
                }
            }
        }
    };

    xhr.setRequestHeader("Content-type", "application/json");
    xhr.send(JSON.stringify(jsonData));
});
```



5537

종말

todoNum	todoContent	todoRdate	todoStatus
1	할일 1	2023-02-20	1
2	할일 2	2023-02-20	3
3	할일 3	2023-02-20	3
4	추가된 일	2023-02-20	1
5	추가된 일2	2023-02-20	1
6	추가된 일3	2023-02-20	1

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

## 4) 기능 구현

### 아이템 삭제시 Delete 처리

```
$(document).on('click', '.del', function(){
    const todoNum = $(this).next().text();

    const content = {"todoNum":todoNum.substring(1)};

    $.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!');
            }
        }
    });
});
```

!-- delete -->

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

Todo

Ready

리스트 삭제 성공!

확인

#1  
할일 1  
2023-02-20

X

#4  
추가된 일  
2023-02-20

X

#7  
추가된 일 5  
2023년 2월 20일

X

#2  
할일 2  
2023-02-20

X

#3  
할일 3  
2023-02-20

X

todoNum	🔑	todoContent	todoRdate	todoStatus
1		할일 1	2023-02-20	1
2		할일 2	2023-02-20	3
3		할일 3	2023-02-20	3
4		추가된 일	2023-02-20	1