13. 통합 구현

내외부 연계 모듈 구현

날짜 : 2023/ 02/ 20 (월)
이름 : 김재준

목차

과정 수행

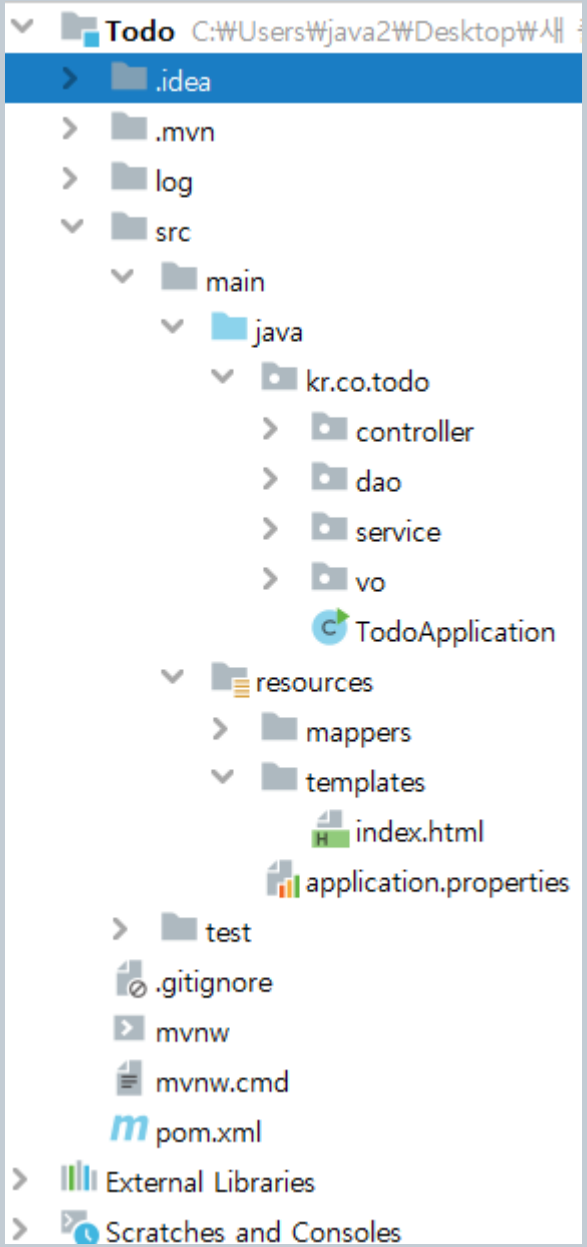
1. 프로젝트 생성

· 프로젝트 : Todo

· Build : Maven

· Language : Java11

· Dependencies : Spring boot, Lombok, Spring Web, Thymeleaf, Mybatis, MySQL



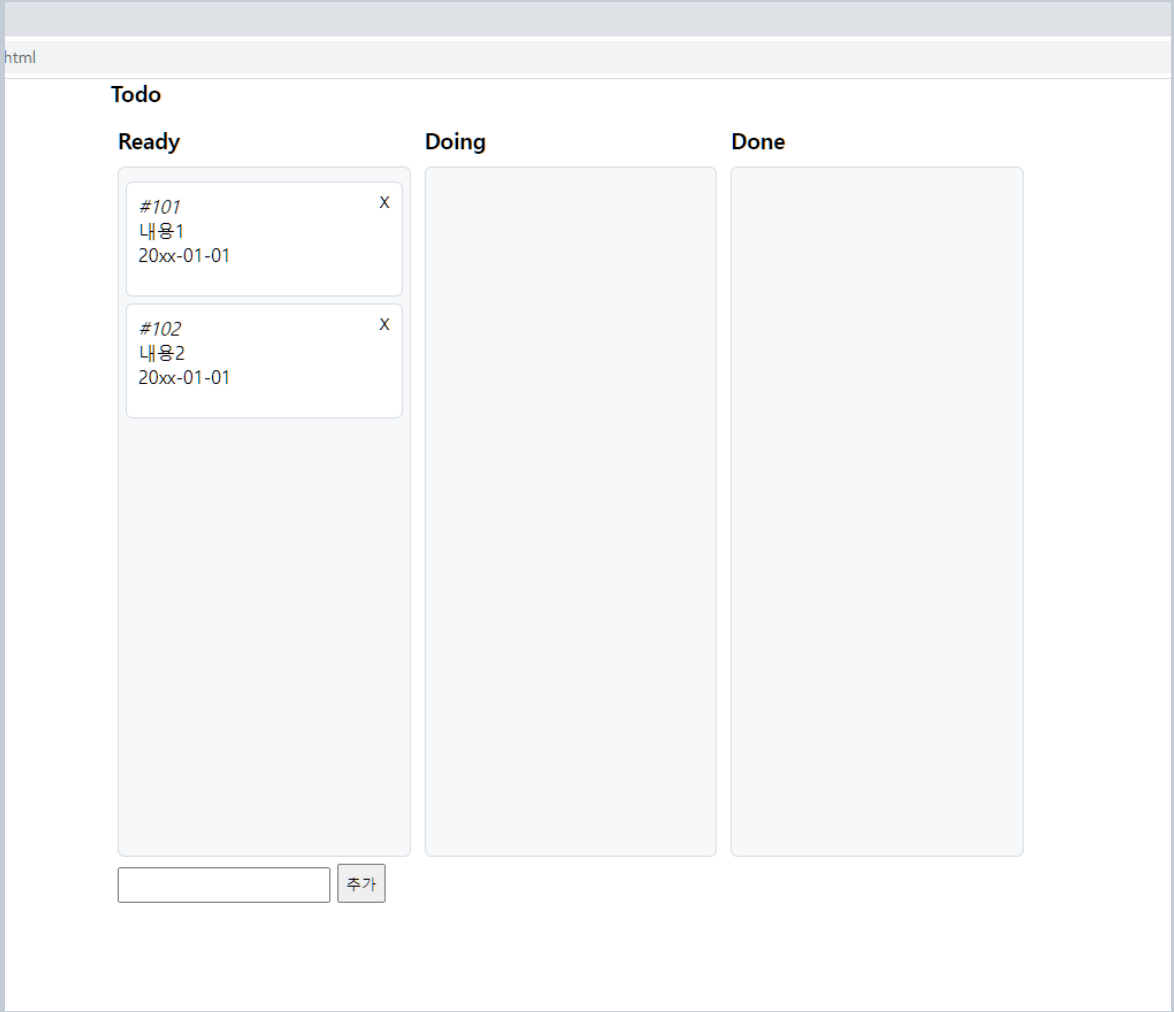
2. 화면구현

· 화면 레이아웃 작업

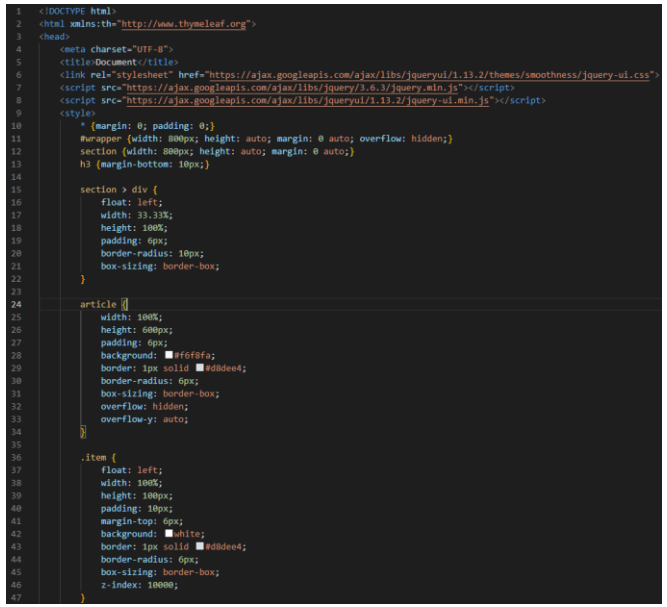
· 아이템 드래깅 처리(jQuery sortable API 활용)

· 하단 리스트 아이템 추가 기능

· 리스트별 아이템 삭제 기능



• 화면 html



```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <link rel="stylesheet" href="https://ajax.googleapis.com/ajax/libs/jqueryui/1.13.2/themes/smoothness/jquery-ui.css">
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.3/jquery.min.js"></script>
    <script src="https://ajax.googleapis.com/ajax/libs/jqueryui/1.13.2/jquery-ui.min.js"></script>
    <style>
        * {margin: 0; padding: 0;}
        #wrapper {width: 800px; height: auto; margin: 0 auto; overflow: hidden;}
        section {width: 800px; height: auto; margin: 0 auto;}
        h3 {margin-bottom: 10px;}

        section > div {
            float: left;
            width: 33.33%;
            height: 100%;
            padding: 6px;
            border-radius: 10px;
            box-sizing: border-box;
        }

        article {
            width: 100%;
            height: 600px;
            padding: 6px;
            background: #f6f8fa;
            border: 1px solid #d8dee4;
            border-radius: 6px;
            box-sizing: border-box;
            overflow: hidden;
            overflow-y: auto;
        }

        .item {
            float: left;
            width: 100%;
            height: 100px;
            padding: 10px;
            margin-top: 6px;
            background: white;
            border: 1px solid #d8dee4;
            border-radius: 6px;
            box-sizing: border-box;
            z-index: 10000;
        }
```

```
        .item > .del {
            float: right;
            background: none;
            border: none;
        }

        .add {
            padding: 6px;
            box-sizing: border-box;
        }

        .add > input {
            padding: 6px;
            box-sizing: border-box;
            outline: none;
        }
    </style>

    <script>
        $(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);
                }
            });

            $('#btnAdd').click(function(){
                let value = $('input[name=todo]').val();
                let item = `<div class='item'>
                                <button class="del">X</button>
                                <em class='tit'>#200</em>
                                <p>내용</p>
                                <span class='date'>2023-03-01</span>
                                </div>`;

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

            $(document).on('click', '.del', function(){
```

```
                $(this).parent().remove();
            });
        });
    </script>
</head>
<body>
    <div id="wrapper">
        <h3>Todo</h3>
        <section>
            <div>
                <h3>Ready</h3>
                <article class="ready" data-status="1">
                    <div class="item" data-no="100">
                        <button class="del">X</button>
                        <em class="tit">#101</em>
                        <p>
                            내용1
                        </p>
                        <span class="date">20xx-01-01</span>
                    </div>
                    <div class="item" data-no="102">
                        <button class="del">X</button>
                        <em class="tit">#102</em>
                        <p>
                            내용2
                        </p>
                        <span class="date">20xx-01-01</span>
                    </div>
                </article>
            </div>
            <div>
                <h3>Doing</h3>
                <article class="doing" data-status="2"></article>
            </div>
            <div>
                <h3>Done</h3>
                <article class="done" data-status="3"></article>
            </div>
        </section>
        <div class="add">
            <input type="text" name="todo"/>
            <input type="button" id="btnAdd" value="추가"/>
        </div>
    </div>
</body>
</html>
```

- 아이템 드래깅

```
<script>
    $(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);
            }
        });
```

- 하단 리스트 아이템 추가

```
        $('#btnAdd').click(function(){
            let value = $('input[name=todo]').val();
            let item = `<div class='item'>
                        <button class="del">X</button>
                        <em class='tit'>#200</em>
                        <p>내용</p>
                        <span class='date'>2023-03-01</span>
                        </div>`;
```

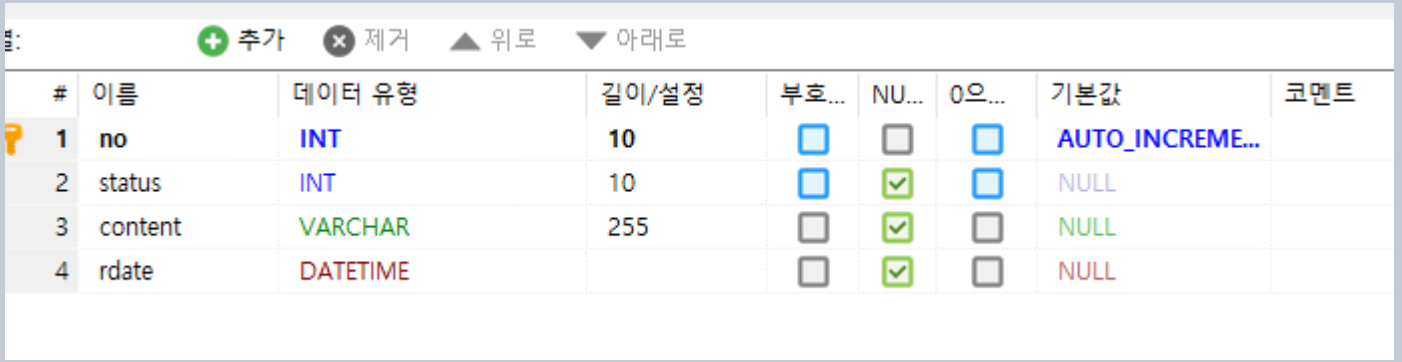
- 리스트별 아이템 삭제

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

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

3. 테이블 설계

· 리스트별 데이터 저장을 위한 테이블 설계

· 테이블명 : Todo



#	이름	데이터 유형	길이/설정	부호...	NU...	0으...	기본값	코멘트
1	no	INT	10	☐	☐	☐	AUTO_INCREME...	
2	status	INT	10	☐	☑	☐	NULL	
3	content	VARCHAR	255	☐	☑	☐	NULL	
4	rdate	DATETIME		☐	☑	☐	NULL	

4. 기능구현

- 메인화면 아이템 select 처리
- 아이템 추가시 Insert 처리
- 아이템 삭제시 Delete 처리
- 아이템 이동시 Update 처리

- VO

```java
package kr.co.todo.vo;

import ...

@Getter
@Setter
@NoArgsConstructor
@AllArgsConstructor
@ToString
public class TodoVO {
    private int no;
    private String content;
    private int status;
    private String rdate;
}
```

- DAO

```java

import ...

@Mapper
@Repository
public interface MainDAO {
    // create

    /**
     * 2023/02/20 // 김재준 // 투두 리스트 생성
     */
    public int insertTodoList(TodoVO vo);

    // read

    /**
     * 2023/02/20 // 김재준 // 투두 리스트 불라오기
     * @return
     */
    public List<TodoVO> selectTodoList();
    // upload

    /**
     * 2023/02/20 // 김재준 // 투두리스트 업데이트
     *      필요한 값
     *      todoNum
     *      todoStatus
     * @param data
     * @return 결과 값
     */
    public int updateTodoList(Map<String, String> data);
    // delete

    /**
     * 2023/02/20 // 김재준 // 투두리스트 삭제
     */
    public int deleteTodoList(TodoVO vo);

}
```

- **Service**

```java
package kr.co.todo.service;

import ...

@Service
public class MainService {

    /**
     * 2023/02/20 // 김재준 // dao 연결
     */
    @Autowired
    private MainDAO dao;

    // create

    /**
     * 2023/02/20 // 김재준 // 투두 리스트 생성
     */
    public int insertTodoList(TodoVO vo) {
        return dao.insertTodoList(vo);
    }

    // read

    /**
     * 2023/02/20 // 김재준 // 투두 리스트 불라오기
     * 리스트 불러온후 상태 값에 따라 분리
     * @return 1, 2, 3
     */
    public Map<Integer, List<TodoVO>> selectTodoList() {
        List<TodoVO> list = dao.selectTodoList();
        return list.stream().collect(Collectors.groupingBy(TodoVO::getStatus));
    }

    // update

    /**
     * 2023/02/20 // 김재준 // 투두리스트 업데이트
     *      필요한 값
     *      todoNum
     *      todoStatus
     * @return 결과 값
     */
    public int updateTodoList(Map<String, String> data) {
        return dao.updateTodoList(data);
    }

    // delete

    /**
     * 2023/02/20 // 김재준 // 투두리스트 삭제
     */
    public int deleteTodoList(TodoVO vo) {
        return dao.deleteTodoList(vo);
    }

    // service
}
```

• 메인화면 아이템 Select 컨트롤러

```java
@GetMapping(value = {"/", "/index"})
public String index(Model model) {
    // 최초 접근시 각 상태의 게시물 불러오기
    Map<Integer, List<TodoVO>> data = service.selectTodoList();

    // 전송
    model.addAttribute( attributeName: "data", data);

    return "index";
}
```

• 아이템 추가시 Insert 컨트롤러

```java
@ResponseBody
@PostMapping("todoInsert")
public Map<String, Object> todoInsert(@RequestBody TodoVO vo) {
    // 리턴 할 값 선언
    int result = 0;

    // 들어오는 값으로 row 생성
    result = service.insertTodoList(vo);

    // 리턴하는 map 생성
    Map<String, Object> resultMap = new HashMap<>();
    resultMap.put("result", result);
    resultMap.put("vo", vo);

    // 리턴
    return resultMap;
}
```

• Insert 스크립트

```javascript
$('#btnAdd').click(function(){

    let value = $('input[name=todo]').val();
    const date = new Date();
    const dateS = date.toLocaleDateString('ko',{
                        year:'numeric',
                        month: '2-digit',
                        day: 'numeric'
                    }).replace(/\./g,'').replace(/\s/g,'-');

    jsondata = {"content":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.no+`</em>
                                    <p>`+value+`</p>
                                    <span class="date">`+dateS+`</span>
                                </div>`;

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

• 아이템 삭제시 Delete 컨트롤러

```java
@ResponseBody
@PostMapping("todoDelete")
public Map<String, Object> todoDelete(@RequestBody TodoVO vo) {
    // 리턴 할 값 선언
    int result = 0;

    // 들어오는 값으로 row 생성
    result = service.deleteTodoList(vo);

    // 리턴하는 map 생성
    Map<String, Object> resultMap = new HashMap<>();
    resultMap.put("result", result);

    // 리턴
    return resultMap;
}
```

• Delete 스크립트

```javascript
$(document).on('click', '.del', function(){
    const no = $(this).next().text();

    const content = {"no":no.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!');
            }
        }
    });
});
```

• 아이템 이동시 Update 컨트롤러

```java
@ResponseBody
@PostMapping("todoUpdate")
public Map<String, Object> todoUpdate(@RequestBody Map<String, String> data) {
    // 리턴 할 값 선언
    int result = 0;

    // 들어오는 값으로 게시물 변화
    result = service.updateTodoList(data);

    // 리턴 하는 map 생성
    Map<String, Object> resultMap = new HashMap<>();
    resultMap.put("result", result);

    // 리턴
    return resultMap;
}
```

• Update 스크립트

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

            const content = {"no":no,"status":status}

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

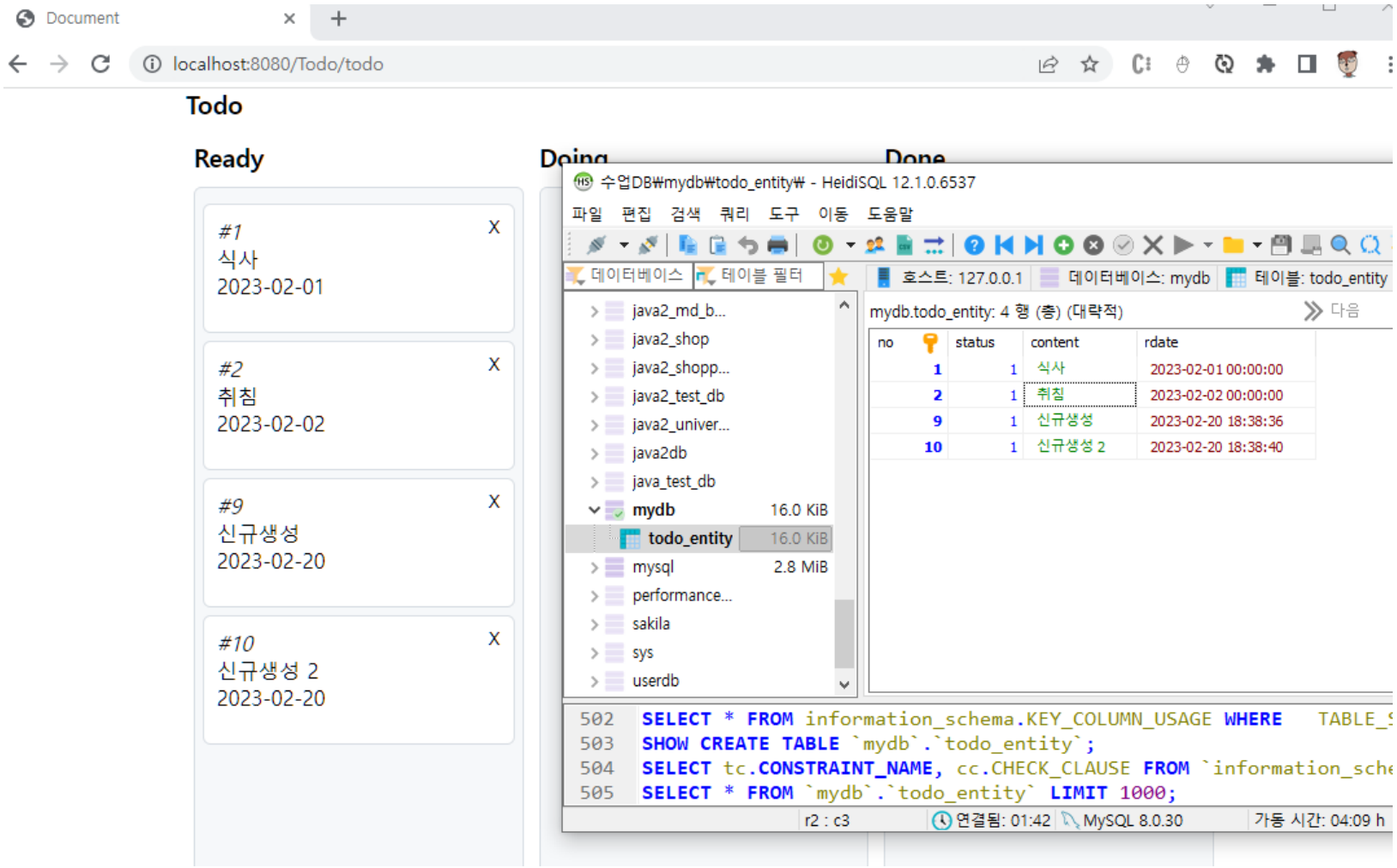
● Mapper

```xml
<?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">
    <!-- create -->
    <insert id="insertTodoList" parameterType="kr.co.todo.vo.TodoVO"
            useGeneratedKeys="true" keyProperty="no" keyColumn="no">
        INSERT INTO todo SET content=#{content} , rdate=NOW();
    </insert>

    <!-- read -->
    <!-- 2023/02/20 // 김철학 // todo리스트 불러오기 -->
    <select id="selectTodoList" resultType="kr.co.todo.vo.TodoVO">
        SELECT * FROM todo;
    </select>

    <!-- upload -->
    <!-- 2023/02/20 // 김철학 // todo리스트 업데이트 쿼리문 -->
    <update id="updateTodoList">
        UPDATE todo SET status=${status} WHERE no=${no};
    </update>

    <!-- delete -->
    <delete id="deleteTodoList" parameterType="kr.co.todo.vo.TodoVO">
        DELETE FROM todo WHERE no=#{no};
    </delete>
</mapper>
```

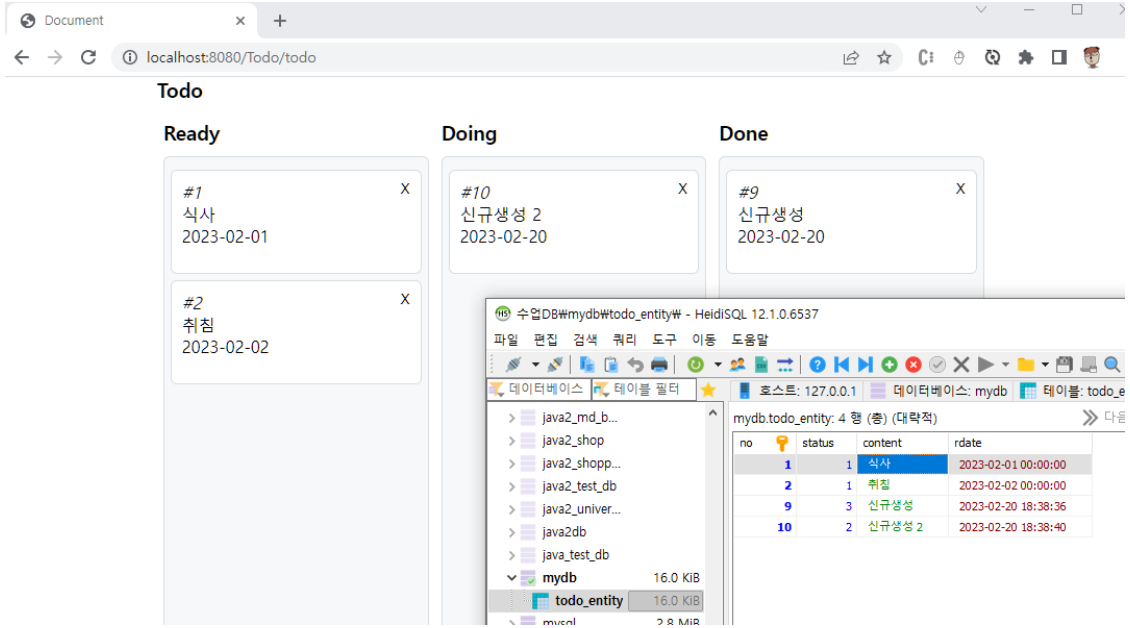
● index

```html
<h3>Todo</h3>
<section>
    <div>
        <h3>Ready</h3>
        <article class="ready" data-status="1">
            <th:block th:each="article:${data[1]}">
                <div class="item" th:data-no="${article.no}">
                    <button class="del">X</button>
                    <em class="tit" th:text="${'#'+ article.no}">#100</em>
                    <p th:text="${article.content}">내용1</p>
                    <span class="date" th:text="${article.rdate}">20xx-01-01</span>
                </div>
            </th:block>
        </article>
    </div>
    <div>
        <h3>Doing</h3>
        <article class="doing" data-status="2">
            <th:block th:each="article:${data[2]}">
                <div class="item" th:data-no="${article.no}">
                    <button class="del">X</button>
                    <em class="tit" th:text="${'#'+ article.no}">#100</em>
                    <p th:text="${article.content}">내용1</p>
                    <span class="date" th:text="${article.rdate}">20xx-01-01</span>
                </div>
            </th:block>
        </article>
    </div>
    <div>
        <h3>Done</h3>
        <article class="done" data-status="3">
            <th:block th:each="article:${data[3]}">
                <div class="item" th:data-no="${article.no}">
                    <button class="del">X</button>
                    <em class="tit" th:text="${'#'+ article.no}">#100</em>
                    <p th:text="${article.content}">내용1</p>
                    <span class="date" th:text="${article.rdate}">20xx-01-01</span>
                </div>
```

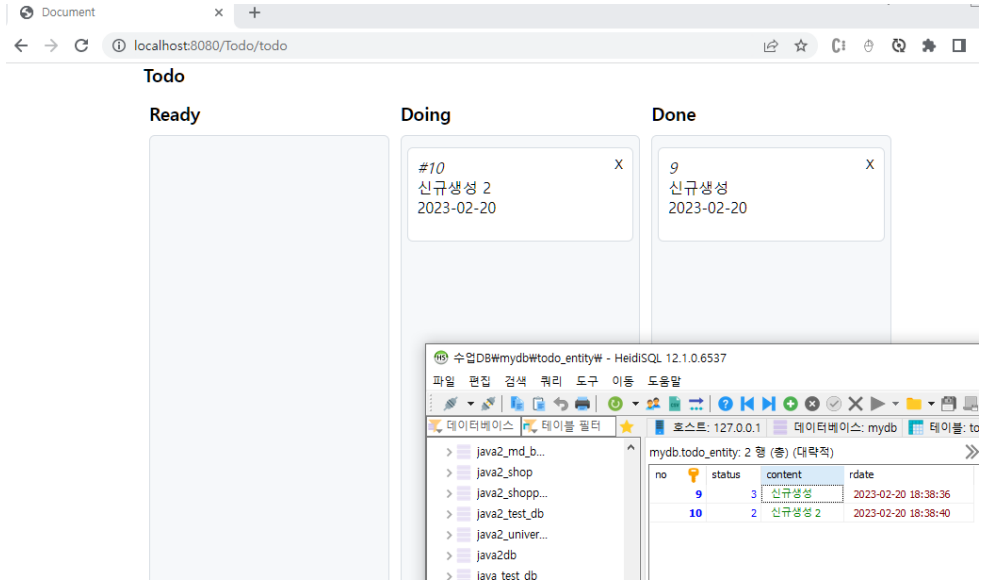
• 신규생성


Document
localhost:8080/Todo/todo
Todo
Ready
Doing
Done
#1
식사
2023-02-01
#2
취침
2023-02-02
#9
신규생성
2023-02-20
#10
신규생성 2
2023-02-20
수업DB₩mydb₩todo_entity₩ - HeidiSQL 12.1.0.6537
파일 편집 검색 쿼리 도구 이동 도움말
데이터베이스
테이블 필터
호스트: 127.0.0.1
데이터베이스: mydb
테이블: todo_entity
mydb.todo_entity: 4 행 (총) (대략적)
다음
no status content rdate
1 1 식사 2023-02-01 00:00:00
2 1 취침 2023-02-02 00:00:00
9 1 신규생성 2023-02-20 18:38:36
10 1 신규생성 2 2023-02-20 18:38:40
java2_md_b...
java2_shop
java2_shopp...
java2_test_db
java2_univer...
java2db
java_test_db
mydb 16.0 KiB
todo_entity 16.0 KiB
mysql 2.8 MiB
performance...
sakila
sys
userdb
SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE TABLE_
SHOW CREATE TABLE `mydb`.`todo_entity`;
SELECT tc.CONSTRAINT_NAME, cc.CHECK_CLAUSE FROM `information_sche
SELECT * FROM `mydb`.`todo_entity` LIMIT 1000;
r2 : c3
연결됨: 01:42
MySQL 8.0.30
가동 시간: 04:09 h

- **status 이동**


Document
localhost:8080/Todo/todo
Todo
Ready
Doing
Done
#1
식사
2023-02-01
#2
취침
2023-02-02
#10
신규생성 2
2023-02-20
#9
신규생성
2023-02-20
수업DB₩mydb₩todo_entity₩ - HeidiSQL 12.1.0.6537
mydb.todo_entity: 4 행 (총) (대략적)
no status content rdate
1 1 식사 2023-02-01 00:00:00
2 1 취침 2023-02-02 00:00:00
9 3 신규생성 2023-02-20 18:38:36
10 2 신규생성 2 2023-02-20 18:38:40

- **삭제**


Document
localhost:8080/Todo/todo
Todo
Ready
Doing
Done
#10
신규생성 2
2023-02-20
9
신규생성
2023-02-20
수업DB₩mydb₩todo_entity₩ - HeidiSQL 12.1.0.6537
mydb.todo_entity: 2 행 (총) (대략적)
no status content rdate
9 3 신규생성 2023-02-20 18:38:36
10 2 신규생성 2 2023-02-20 18:38:40