

❖ 오류내용과 이유(2nd 프젝)

1. 주문목록의 제목에 a태그를 넣을경우 제목들이 사라진다.

```
<div class="content_name"><a href="/orderProductUpdateForm.jsp?ordernum=<%=rs.getInt("ORDER_NUM") %>"><%= rs.getString("ITEM_NAME") %></a></div>
```

- 이유: css의 a 태그의 값을 0으로 해주었기 때문에 안나왔다

2. orderProductUpdateForm 에서 디비에 저장된 목록값이 나오지 않는다

- 이유: int 의 초기값이 없었기 때문에 꼭 초기값을 0 또는 null 을 넣어줘야한다. 변수 int 대신 클래스 Integer를 넣어줄 수 있다
클래스 Integer를 넣어주는게 나중에 활용이 좋다

```
String itemsize = "";//사이즈  
Integer quantity = null;//수량
```

3. posted content type isn't multipart/form-data 오류

- 이유: order_number의 값을 넣어주지 않았다. ordernumber 로 string 값을 넣고 나중에 int로 바꿔줬다

4. 자바스크립트에서 for 문으로 배열돌릴때 오류, btn[i] 로 [i] 값을 똑같이 넣어줘야한다.

```
let btn = Array.from(document.querySelectorAll(".status"));  
let menu = document.querySelector("#nav")
```

```
    console.log('btn:', btn);  
    for(let i=0; i < btn.length; i++){  
  
        btn[i].addEventListener("dblclick",()=>{  
            btn[i].classList.add("active");  
            nav.classList.add("active");  
        });  
  
        btn[i].addEventListener("click",()=>{  
            btn[i].classList.remove("active");  
            nav.classList.remove("active");  
        });  
    }  
}
```

5. update_status.jsp ////오류내용: Cannot parse null string

오류: Cannot invoke "java.sql.PreparedStatement.setString(int, String)" because "pstmt" is null

- pstmt.executeUpdate();를 update_status.jsp에서 빠트렸다

6. orderList.jsp 에서 content-info-box 를 위에서 가져온 ordernum와 content-info-box의 ordernum가 일치할 때 반복문으로 코멘트 내용을 가져오기

- 자바스크립트에 새로운 ordernum의 변수를 만들어주고 전에 만들어진 ordernum의 변수와 새로 만들어진 변수가 같을 때 content-info-box를 표시하고 같지 않을 땐 표시를 해주지 않는다.

오류: order.js:44 Uncaught TypeError: Cannot set properties of undefined (setting 'selected')

at HTMLDivElement.<anonymous> (order.js:44:57)

오류: For input string: "null"

오류: Uncaught TypeError: Cannot read properties of null (reading 'innerText')

at order.js:49:61

at NodeList.forEach (<anonymous>)

at HTMLDivElement.<anonymous> (order.js:48:26)

이런 오류들이 생겼을땐, 값을 제대로 출력할 수 있는지 콘솔에 하나씩 찍으면서 오류를 해결해 나갈 수 있다.//console.log("forEach box: ', box)

```
var orderNum = btn1[i].parentElement.querySelector(".ordernum").innerText; - 먼저 선언된 orderNum
```

```
contentInfoBoxes.forEach((box) => {  
    //console.log("forEach box: ', box);  
    var boxOrderNum = box.querySelector(".ordernum").innerText;  
    //console.log("boxOrderNum: ', boxOrderNum);  
  
    if (boxOrderNum === orderNum) {  
        // 주문 번호가 일치하면 해당 content-info-box를 보여줌  
        box.style.display = "block";  
    } else {  
        // 주문 번호가 일치하지 않으면 해당 content-info-box를 숨김  
        box.style.display = "none";  
    }  
});
```

6. css 자바스크립트가 안먹힌다

```
document.getElementById('toggleButton').addEventListener('click', toggleWidth);

function toggleWidth() {
  const nav = document.getElementById('nav');
  const followWrap = document.querySelector('#nav .follow-wrap');
  const fcontent = document.getElementById('fcontent');
  const commenter = document.querySelector('#nav .content-info-box .commenter');

  if (nav.style.width === '1000px') {
    nav.style.width = '400px'; // 변경할 너비 설정
    followWrap.style.width = '430px';
    fcontent.cols = 58;
    commenter.classList.add('position-right-36'); // 클래스 추가

  } else {
    nav.style.width = '1000px'; // 초기 너비로 복원
    followWrap.style.width = '1010px';
    fcontent.cols = 100;
    commenter.classList.remove('position-right-36'); // 클래스 제거
  }
}
```

이유:

```
<nav id="nav" class="active" style="width: 400px;">
  <div class="wrap">
    <div class="orderInfo">...</div>
    <div class="content-info">...</div> flex
    <div class="content-info-box" style="display: none;">...</div> == $0
    <div class="content-info-box" style="display: block;">...</div>
    <div class="content-info-box" style="display: none;">...</div>
    <div class="content-info-box" style="display: none;">...</div>
    <div class="content-info-box" style="display: none;">...</div>
    <div class="content-info-box" style="display: none;">...</div>
    <div class="content-info-box" style="display: none;">...</div>
    <div class="content-info-box" style="display: block;">...</div>
```

`document.querySelector` 는 하나만, 첫번째만 가져오는 코드로 **All**로 바꿔줘야한다.
값이 여러개이기 때문에 **for**문으로 돌려서 여러 값을 가져올 수 있다.

```
for ( var i = 0; i < commenter.length; i++ ) {
  commenter[i].style.right = '36%';
}
```

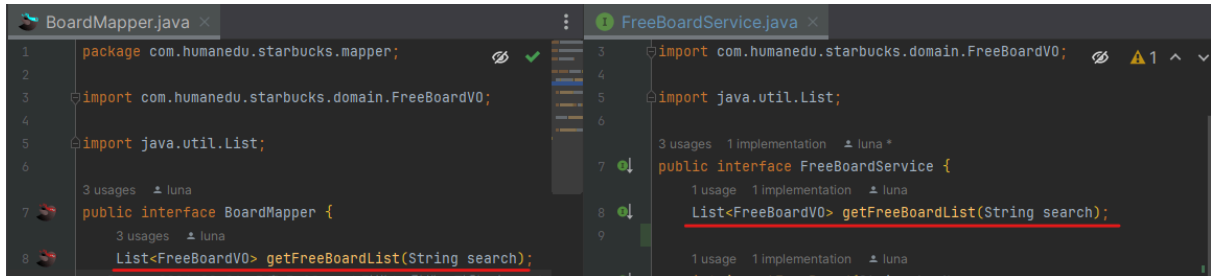
참고자료

<https://www.codingfactory.net/10410> <https://www.codingfactory.net/10413>

❖ 스벅 스프링 다시 복습

질문

1. interface mapper 와 service 메서드이름을 달리해야하는데 왜 아래는 같은가?



- 다르게 해도 된다. 적당한 다른 이름이 없고 메소드 기능이 거의 똑같아서 같은 이름으로 만들었다

2. BoardMapper / BoardService 에서 int 로 주는 이유?

- int 말고 void 로는 쓸 수 없나? int, void 말고 다른 것도 있나?

// 쓸 수 있다

3. BoardServiceImpl에서 if(!search.equals("")) 사용

@Override

```
public List<FreeBoardVO> getFreeBoardList(String search) {  
    List<FreeBoardVO> freeBoardVOList = null;  
    if(search == null) {  
        freeBoardVOList = boardMapper.getFreeBoardList(search);  
    } else {  
        if (!search.equals("")) {  
            freeBoardVOList = boardMapper.getFreeBoardList(search);  
        }  
    }  
    return freeBoardVOList;  
}
```

search 가 널 일 경우 / search가 널이 아닌 경우 모두 boardMapper.xml에서 실행해준다.

WHERE

1=1 // true라고 쓰면 에러가 나서 1=1이라고 사용한다

```
<if test="search != null">
```

```
<if test="search != "">
```

```
    AND SUBJECT LIKE '%' || #{search} || '%'
```

```
</if>
```

```
<if test="search == "">
```

```
    AND 1=0 // false 라서 sql 문 실행을 하지않아서 아무것도 나오지않는다
```

```
</if>
```

```
</if>
```

```
ORDER BY
```

NUM DESC fetch first 5 rows only

널일 경우, if문이 실행되고 아닌경우 실행되지 않는다.

`<if test="search == "">` `AND 1=0` `</if>` `/// search가 공백일경우 1=0 으로 false? 어떻게 모든 리스트가 보이는가?`

1. adminNoticeList 설정하기

DBConfig – 중요하다


```
no usages
@Configuration
@MapperScan(basePackages = {"com.humanedu.starbucks.mapper"})
```

<https://shorturl.at/acqCD>

xml에서 `<mapper>` 태그와 `<select>`태그의 값 주의

mapper - 연결되는 주소

select - 디비 테이블의 변수값이 저장 된 곳



```
BoardMapper.xml x BoardMapper.java x
<?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="com.humanedu.starbucks.mapper.BoardMapper">
    <select id="getFreeBoardList" parameterType="String" resultType="com.humanedu.starbucks.domain.FreeBoardVO">
```

오라클에서 사용하는 문법이 스프링에서는 다르다

```
<select id="getFreeBoardList" resultType="com.humanedu.starbucks.domain.FreeBoardVO">
    SELECT
        NUM,
        SUBJECT,
        HIT,
        REGDATE
    FROM
        BO_FREE
    WHERE
        SUBJECT LIKE '%' + searchText + '%'
    ORDER BY
        NUM DESC fetch first 5 rows only
</select>
```

```

<select id="getFreeBoardList" parameterType="String" resultType="com.human
SELECT
    NUM,
    SUBJECT,
    HIT,
    REGDATE
FROM
    BO_FREE
WHERE
    1=1
    <if test = "search != null and search != ''">
        AND SUBJECT LIKE '%' || ${search} || '%'
    </if>
ORDER BY
    NUM DESC fetch first 5 rows only
</select>

```

- **DBconfig** 설정 잘 해놓기

아래코드는 mybatis와 연결시키는 코드설정

```

sqlSessionFactory.setMapperLocations(new
PathMatchingResourcePatternResolver().getResources("classpath:mybatis/**/*.xml
1"));

```

2. adminNoticeInsert 설정하기

korname이 아니라 korName?

자바 변수명으로 name 값을 korName으로 바꾸주었다.

파일은 <List>로 준다

3. adminNoticedelete 설정 > @RequestMapping으로

❖ 2st 프젝 - JSP 를 스프링으로 전환

1. 수정하려는데 num 값이 넘어오지 않는다

```
no usages
@GetMapping("/orderProductUpdateForm")
public String orderProductUpdateForm(@RequestParam("num") int num, Model model){
    OrderVO orderVO = orderService.getOneList(num);
    ✓ model.addAttribute(attributeName: "order", orderVO);

    return "orderProductUpdateForm";
}
```

컨트롤러에서 디버깅 후 orderVO 값을 보면 num=0 값으로 처리된다 이유는 getOneList를 따라가면 mapper.xml 에 있는 getListOne에 num을 받아주는 컬럼값이 없었다. ORDER_NUM을 넣어주니 num 값이 넣어진다.

```
<select id="getListOne" resultType="c
    SELECT
        ORDER_NUM,
        REQUESTER,
        BRAND_NAME,
        ITEM_TYPE,
        ORDER_PLACE,
        ITEM_NAME,
        ITEM_SIZE,
        QUANTITY,
        CHEST,
        FRONT_LENGTH,
        SLEEVE_LENGTH,
        BACK_WIDTH,
        ARM_WIDTH,
        REQUIREMENTS,
        PRODUCT_DETAILS,
        FILE_PATH
    FROM ORDER_PRODUCT
    WHERE ORDER_NUM = #{num}
</select>
```

where에서만 선언하는것이 아니라 select문 에서도 선언해줘야한다.

2. 검색기능 구현하기

<!-- 제품 이름 검색 -->

```
<input type="search" name="search-text" id="search-text" placeholder="제품이름을 검색하세요."
value='<c:out value="\${search}" />'><a class="search" href="javascript: searchText();">
<span class="material-symbols-outlined" id="search">search</span></a>
```

검색할 수 있는 내용 - 의뢰자, 의뢰처, 제품 이름, 날짜
배운 ex02

```
<form id='searchForm' action='./orderList' method='get'>
<select name='type' />
<option value="">'search'</option>
<option value="W" <c:out value="\${criteria.type == 'W' ? 'selected' : ''}" />>작성자</option>
<option value="O" <c:if test="\${criteria.type == 'O'}">selected</c:if>>의뢰처</option>
<option value="P" <c:out value="\${criteria.type == 'P' ? 'selected' : ''}" />>제품</option>
</select>
<input type='text' name='keyword' value='<c:out value="\${criteria.keyword}' />' />
<%-- <input type='hidden' name='pageNum' />
<input type='hidden' name='amount' /> --%>
<span class="material-symbols-outlined" id="search">search</span></a>
</form>
```

```
//<button class='btn btn-primary'>Search</button>
```

2. 날짜검색 구현하기

<https://baessi.tistory.com/185>

mapper.xml에 아래 추가

```
<if test="startDt != null and startDt != ''">
    <![CDATA[
        or ORDER_DATE >= #{startDt}
    ]]>
</if>
<if test="endDt != null and endDt != ''">
    <![CDATA[
        and ORDER_DATE <= #{endDt}
    ]]>
</if>
```


3. 날짜 & 검색 구현

```
<select id="getOrderList" parameterType="String"
resultType="com.humanedu.firstproject.domain.OrderVO"
resultMap="mapping_orderlist">
    <![CDATA[
        SELECT *
        FROM (SELECT
            ORDER_NUM, STATUS, TO_CHAR(ORDER_DATE, 'YYYY-MM-DD') AS
ORDER_DATE, ITEM_NAME, QUANTITY, REQUESTER, ORDER_PLACE
            FROM ORDER_PRODUCT)
    ]]>
    <trim prefix="where (" suffix=)" prefixOverrides="or">
        <foreach item="type" collection="typeArr">
            <if test="type == 'W'.toString()">
                or REQUESTER like '%' || #{keyword} || '%'
            </if>
            <if test="type == 'O'.toString()">
                or ORDER_PLACE like '%' || #{keyword} || '%'
            </if>
            <if test="type == 'N'.toString()">
                or ITEM_NAME like '%' || #{keyword} || '%'
            </if>
            <if test="startDt != null and startDt != ''">
                <![CDATA[
                    or ORDER_DATE >= #{startDt}
                ]]>
            </if>
            <if test="endDt != null and endDt != ''">
                <![CDATA[
                    and ORDER_DATE <= #{endDt}
                ]]>
            </if>
        </foreach>
    </trim>
    ORDER BY ORDER_DATE DESC fetch first 5 rows only
</select>
```

4. 페이징 처리

오류: **mapper.xml**

```
<select id="getListWithPaging"
resultType="com.humanedu.firstproject.domain.OrderVO"
resultMap="mapping_orderlist">
    <![CDATA[
SELECT
    *
FROM (
    SELECT /*+INDEX_DESC (ORDER_PRODUCT pk_ORDER_NUM) */
        ROWNUM rn, ORDER_NUM, STATUS, TO_CHAR(ORDER_DATE, 'YYYY-MM-DD') AS
ORDER_DATE, ITEM_NAME, QUANTITY, REQUESTER, ORDER_PLACE
    FROM
        ORDER_PRODUCT
    WHERE ROWNUM <= #{pageNum} * #{amount}
    ORDER BY ORDER_DATE DESC
    )
]]>
    <trim prefix="where (" suffix=")" prefixOverrides="or">
        <foreach item="type" collection="typeArr">
            <if test="type == 'W'.toString()">
                or REQUESTER like '%' || #{keyword} || '%'
            </if>
            <if test="type == 'O'.toString()">
                or ORDER_PLACE like '%' || #{keyword} || '%'
            </if>
            <if test="type == 'N'.toString()">
                or ITEM_NAME like '%' || #{keyword} || '%'
            </if>
            <if test="startDt != null and startDt != ''">
                <![CDATA[
                and ORDER_DATE >= #{startDt}
                ]]>
            </if>
            <if test="endDt != null and endDt != ''">
                <![CDATA[
                and ORDER_DATE <= #{endDt}
                ]]>
            </if>
            and <![CDATA[ rn > (#{pageNum} - 1) * #{amount} ]]>
        </foreach>
    </trim>
</select>
```

해결순서1: 오류난 sql문을Dbeaver에서 확인해봐야한다

```
SELECT
*
FROM
(
SELECT
/*+INDEX_DESC (ORDER_PRODUCT pk_ORDER_NUM) */
ROWNUM rn,
ORDER_NUM,
STATUS,
TO_CHAR (ORDER_DATE, 'YYYY-MM-DD') AS ORDER_DATE,
ITEM_NAME,
QUANTITY,
REQUESTER,
ORDER_PLACE
FROM
ORDER_PRODUCT
WHERE 1=1
ORDER BY ORDER_DATE DESC
)
WHERE 1=1
AND ROWNUM <= 1 * 5
--ORDER BY
-- ORDER_DATE DESC
;
SELECT
*
FROM
(
SELECT
/*+INDEX_DESC (ORDER_PRODUCT pk_ORDER_NUM) */
ROWNUM rn,
ORDER_NUM,
STATUS,
TO_CHAR (ORDER_DATE, 'YYYY-MM-DD') AS ORDER_DATE,
ITEM_NAME,
QUANTITY,
REQUESTER,
ORDER_PLACE
FROM
ORDER_PRODUCT
WHERE
1 = 1
ORDER BY
ORDER_DATE DESC )
WHERE
(
ORDER_DATE >= '2024-03-12'
AND ORDER_DATE <= '2024-03-12'
AND rn > (1 - 1) * 5 )
;
```

수정된 **mapper.xml**

해결순서**2**: 근데 페이지 **2**번 클릭시 **10**개의 목록으로 보인다

```
<select id="getListWithPaging"
resultType="com.humaneu.firstproject.domain.OrderVO"
resultMap="mapping_orderlist">
    <![CDATA[
        SELECT
            *
        FROM (
            SELECT /*+INDEX_DESC(ORDER_PRODUCT pk_ORDER_NUM) */
                ROWNUM rn, ORDER_NUM, STATUS, TO_CHAR(ORDER_DATE, 'YYYY-MM-DD')
            AS ORDER_DATE, ITEM_NAME, QUANTITY, REQUESTER, ORDER_PLACE
            FROM
                ORDER_PRODUCT
            WHERE 1=1
                ORDER BY ORDER_DATE DESC
        )
    ]]>
    <choose>
        <when test="(type != null and type != ''.toString()) or (startDt !=
null and startDt != '') or (endDt != null and endDt != '') or (keyword !=
null and keyword != '')">
            <trim prefix="where (" suffix=")" prefixOverrides="and | or">
                <foreach item="type" collection="typeArr">
                    <if test="type == 'W'.toString()">
                        REQUESTER like '%' || #{keyword} || '%'
                    </if>
                    <if test="type == 'O'.toString()">
                        ORDER_PLACE like '%' || #{keyword} || '%'
                    </if>
                    <if test="type == 'N'.toString()">
                        ITEM_NAME like '%' || #{keyword} || '%'
                    </if>
                    <if test="startDt != null and startDt != ''">
                        <![CDATA[ and ORDER_DATE >= #{startDt} ]]>
                    </if>
                    <if test="endDt != null and endDt != ''">
                        <![CDATA[ and ORDER_DATE <= #{endDt} ]]>
                    </if>
                    and <![CDATA[ rn > (#{pageNum} - 1) * #{amount} ]]>
                </foreach>
            </trim>
        </when>
        <otherwise>
            <![CDATA[
                WHERE 1=1
                    AND ROWNUM <= #{pageNum} * #{amount}
            ]]>
        </otherwise>
    </choose>
</select>
```

해결순서3:

페이지 번호에 맞게 5개씩 날짜 순서대로 나오지만, 검색 결과가 다 나오지 않는다.

```
<select id="getListWithPaging"
resultType="com.humanedu.firstproject.domain.OrderVO"
resultMap="mapping_orderlist">
    <![CDATA[
        SELECT
            *
        FROM (
            SELECT /*+INDEX_DESC(ORDER_PRODUCT pk_ORDER_NUM) */
                ROWNUM rn, ORDER_NUM, STATUS, TO_CHAR(ORDER_DATE,
'YYYY-MM-DD') AS ORDER_DATE, ITEM_NAME, QUANTITY, REQUESTER, ORDER_PLACE
            FROM
                ( SELECT *
                  FROM ORDER_PRODUCT
                 ORDER BY ORDER_DATE DESC
                )
            )
        ]]>
    <choose>
        <when test="(type != null and type != ''.toString()) or (startDt !=
null and startDt != '') or (endDt != null and endDt != '') or (keyword !=
null and keyword != '')">
            <trim prefix="where (" suffix=")" prefixOverrides="and | or">
                <foreach item="type" collection="typeArr">
                    <if test="type == 'W'.toString()">
                        REQUESTER like '%' || #{keyword} || '%'
                    </if>
                    <if test="type == 'O'.toString()">
                        ORDER_PLACE like '%' || #{keyword} || '%'
                    </if>
                    <if test="type == 'N'.toString()">
                        ITEM_NAME like '%' || #{keyword} || '%'
                    </if>
                    <if test="startDt != null and startDt != ''">
                        <![CDATA[ and ORDER_DATE >= #{startDt} ]]>
                    </if>
                    <if test="endDt != null and endDt != ''">
                        <![CDATA[ and ORDER_DATE <= #{endDt}   ]]>
                    </if>
                    <if test="pageNum != null and pageNum != ''">
                        <![CDATA[
                            and RN <= #{pageNum} * #{amount}
                            and RN > (#{pageNum} - 1) * #{amount}
                        ]]>
                    </if>
                </foreach>
            </trim>
        </when>
        <otherwise>
            <![CDATA[
                WHERE 1=1
                AND RN <= #{pageNum} * #{amount}
            ]]>
        </otherwise>
    </choose>
</select>
```

```

        AND RN > ({pageNum} - 1) * #{amount}

    ]]>
</otherwise>
</choose>
</select>

```

```

SELECT
    *
FROM
    (
        SELECT
            /*+INDEX_DESC (ORDER_PRODUCT pk_ORDER_NUM) */
            ROWNUM rn,
            ORDER_NUM,
            STATUS,
            TO_CHAR (ORDER_DATE, 'YYYY-MM-DD') AS ORDER_DATE,
            ITEM_NAME,
            QUANTITY,
            REQUESTER,
            ORDER_PLACE
        FROM
            (
                SELECT
                    *
                FROM
                    ORDER_PRODUCT
                ORDER BY
                    ORDER_DATE DESC )

        WHERE ORDER_PLACE LIKE '%' || '중국' || '%'
    )
WHERE
    RN <= 1 * 5
    AND RN > (1 - 1) * 5
;

```

해결순서4: 주문목록의 총 갯수 설정해서 한 번더 확인해준다 **getOrderTotal**

```
<!--총 갯수-->
<select id="getOrderTotal" resultType="int">
    SELECT COUNT(*)
    FROM (
        SELECT /*+INDEX_DESC(ORDER_PRODUCT pk_ORDER_NUM) */
        ROWNUM rn, ORDER_NUM, STATUS, TO_CHAR(ORDER_DATE,
'YYYY-MM-DD') AS ORDER_DATE, ITEM_NAME, QUANTITY, REQUESTER, ORDER_PLACE
        FROM ORDER_PRODUCT
        ORDER BY ORDER_DATE DESC
    )
    <trim prefix="WHERE" prefixOverrides="AND | OR">
        <if test="(type != null and type != '').toString()" or (startDt != null
and startDt != '') or (endDt != null and endDt != '') or (keyword != null and
keyword != '')">
            <foreach item="type" collection="typeArr">
                <if test="type == 'W'.toString()">
                    AND REQUESTER like '%' || #{keyword} || '%'
                </if>
                <if test="type == 'O'.toString()">
                    AND ORDER_PLACE like '%' || #{keyword} || '%'
                </if>
                <if test="type == 'N'.toString()">
                    AND ITEM_NAME like '%' || #{keyword} || '%'
                </if>
            </foreach>
            <if test="startDt != null and startDt != ''">
                <![CDATA[ AND ORDER_DATE >= #{startDt} ]]>
            </if>
            <if test="endDt != null and endDt != ''">
                <![CDATA[ AND ORDER_DATE <= #{endDt} ]]>
            </if>
        </if>
    </trim>
    ORDER BY ORDER_DATE DESC
</select>
```

오류이유: **ROWNUM** 특별함

수정된 **OrderMapper.xml**

<https://dataonair.or.kr/db-tech-reference/d-lounge/technical-data/?mod=document&uid=235192>

```
<select id="getListWithPaging"
resultType="com.humaneu.firstproject.domain.OrderVO"
resultMap="mapping_orderlist">
    SELECT
        *
    FROM (
        SELECT /*+INDEX_DESC (ORDER_PRODUCT pk_ORDER_NUM) */
            ROWNUM rn, ORDER_NUM, STATUS, TO_CHAR(ORDER_DATE, 'YYYY-MM-DD')
        AS ORDER_DATE, ITEM_NAME, QUANTITY, REQUESTER, ORDER_PLACE
        FROM
            ( SELECT *
              FROM ORDER_PRODUCT
              ORDER BY ORDER_DATE DESC
            )

        <choose>
            <when test="(type != null and type != ''.toString()) or (startDt !=
null and startDt != '') or (endDt != null and endDt != '') or (keyword !=
null and keyword != '')">
                <trim prefix="where (" suffix=")" prefixOverrides="and | or">
                    <foreach item="type" collection="typeArr">
                        <if test="type == 'W'.toString()">
                            REQUESTER like '%' || #{keyword} || '%'
                        </if>
                        <if test="type == 'O'.toString()">
                            ORDER_PLACE like '%' || #{keyword} || '%'
                        </if>
                        <if test="type == 'N'.toString()">
                            ITEM_NAME like '%' || #{keyword} || '%'
                        </if>
                        <if test="startDt != null and startDt != ''">
                            <![CDATA[ and ORDER_DATE >= #{startDt} ]]>
                        </if>
                        <if test="endDt != null and endDt != ''">
                            <![CDATA[ and ORDER_DATE <= #{endDt} ]]>
                        </if>
                    </foreach>
                </trim>
            </when>
        </choose>
    )
    <![CDATA[
        WHERE 1=1
            AND RN <= #{pageNum} * #{amount}
            AND RN > (#{pageNum} - 1) * #{amount}
    ]]>
</select>
```


5. #num 값의 오류 `if(!$('#num').val()) // $('#num').val()`

SweetAlert2 사용시 **#num**값이 있어서 다른 두개만 넣어도 **submit**가 된다.

이유

order.js `document.getElementById("num").value = orderNum;` 에서 넘겨준 **num** 값이 `<input type="text" name="num" id="num" value="" style="display: none;" />` 있기 때문에 **#num**값을 넣지 않아도 있다.

해결

`if(!$('#num').val())` 를 삭제하고 **#commenter** 과 **#content** 만 조건문으로 넣어주고 **#num** 은 자바스크립트로

```
var orderNum2 = btn1[i].parentElement.querySelector(".ordernum").innerText.trim();
document.getElementById("num2").value = orderNum2;
```

값을 주문번호에 넣어 클릭시 자동으로 번호가 나올 수 있게 한다

```
<input type="text" name="num" id="num2">
```

일반js

```
function prevOkTextBox() {
    if(!$('#num').val()) {
        alert('주문번호를 입력하세요.');

$('#num').focus();



return;



}



if(!$('#commenter').val()) {



alert('작성자를 입력하세요.');



$('#commenter').focus();



return;



}



if(!$('#content').val()) {



alert('내용을 입력하세요.');



$('#content').focus();



return;



}



if (confirm('게시후 삭제할 수 없습니다. 정말 게시하겠습니까?')) {



document.getElementById('form2').submit()



}



}


```

일반sweetalert2

```
function checkAllFields() {
    return ($('#num').val() && $('#commenter').val() && $('#content').val());
}
function prevOkTextBox() {
    //console.log($('#num').val()); 여기에 찍어서 나오는지 확인해본다.
```

```
if (!checkAllFields()) {  
    Swal.fire({  
        title: '입력 오류',  
        text: '모든 정보를 입력하세요.',  
        icon: 'error',  
        confirmButtonColor: '#48088A'  
    });  
    return;  
}
```

<https://birdie-hyun.tistory.com/31>

<https://inpa.tistory.com/entry/SweetAlert2-%F0%9F%93%9A-%EC%84%A4%EC%B9%98-%EC%82%AC%EC%9A%A9#sweetalert2>

6. 외래키오류 - 오라클 **FK** 오류 삭제를 **primary** 와 **fk** 둘다 해주어야한다

Mapper.xml 에서 수정

mybatis begin end transaction 검색 // **transaction** 중요

<https://kindloveit.tistory.com/26> 해결방안

❖ 파이널 프젝 오류내용과 이유

1. mybatis의 Mapper.xml 의 오류

```
<select id="getItemInfoList"
resultType="com.cnr_furniture.domain.ItemInfo.ItemInfoVO">
    SELECT ROWNUM rn, I_ID, I_NAME, I_COLOR, I_STANDARD, I_TYPE, I_USES
    FROM ITEM
    where 1=1
    <foreach item="type" collection="typeArr">
        <if test="type == 'INum'.toString()">
            and I_ID like '%' || #{keyword} || '%'
        </if>
        <if test="type == 'IName'.toString()">
            and I_NAME like '%' || #{keyword} || '%'
        </if>
        <if test="type == 'IUses'.toString()">
            and I_USES like '%' || #{keyword} || '%'
        </if>
    </foreach>
    ORDER BY I_ID ASC
</select>
```

select 구문에서 <foreach> 구문이 무시되고 모든 값을 출력한다 디버그를 했을때도 입력한 값은 keyword로 찍히지만 모든 값을 출력한다.

이유: Criteria(ItemInfoSearch DTO) 에 있는 값

```
public String[] getTypeArr() {
    return type == null ? new String[]{} : type.split("");
}
```

- 1) Array사용할 때 split 를 사용해 type이 INum일 때 I, N, u, m 으로 출력된다
- 2) 그러므로 xml 에 사용한 collection="typeArr" 이 제대로 작동하지 않는다.

- 3) **java String 배열 초기화 방법을 검색 >**

<https://codechacha.com/ko/java-init-string-array/>

- 4) split 가 아닌 배열을 초기화하고 값을 할당해줘야한다

```
public String[] getTypeArr() {
    return type == null ? new String[] {} : new String[] {type};
}
```

- 5) mapper.xml에서는 getTypeArr를 쓰지 않고 typeArr로 사용한다.

2. bom.jsp에서 제품번호를 클릭하면 자재정보 출력하기

콘솔에는 클릭된 제품번호의 자재정보들이 출력되지만 바로 에러가 뜬다 – **too many getOneList.**

이유> 너무많은 getOneList가 있다. 확인 해보니 한개 값만 가져오게 MVC mapper, service에 설정해 놓았다.

```
2 usages new * 1 related problem
BomVO getListBom(int i_id); // 제품 id 하나
```

하나를 가져오는 것이 아니기 때문에 List로 전체 수정 해주기

```
2 usages new *
List<BomVO> getListBom(int i_id); // 제품의 bom 리스트
```

<https://baessi.tistory.com/144>

varStatus = "status" 를 사용해 인덱스값을 출력해볼 수 있다

```
<tbody class="table-content checkBomList">
  <!-- <c:forEach var="Bom" items="${bomList}" varStatus="status">
    <tr>
      <td id="rn"><c:out value="${status.index}" /><c:out value="${Bom.rn}" /></td>
      <td id="b_material_id"><c:out value="${status.index}" /><c:out value="${Bom.b_material_id}" /></td>
      <td id="m_name"><c:out value="${status.index}" /><c:out value="${Bom.m_name}" /></td>
      <td id="b_unit"><c:out value="${status.index}" /><c:out value="${Bom.b_unit}" /></td>
      <td id="b_material_quantity"><c:out value="${status.index}" /><c:out value="${Bom.b_material_quantit
    </tr>
  </c:forEach> --%>
</tbody>
```

자바스크립트를 사용해서 bomList가 출력되는지 확인해본다

```
/* Bom 목록의 제품번호 값과 일치하면 Bom세부목록에 Bom내용 불러오기 */
function bomList(i_id) {
  $.get("/bomList/"+i_id, function(result) {
    console.log("bomList:", result);

    /* $('#b_material_id').val(result.b_material_id);
    $('#m_name').val(result.m_name);
    $('#b_unit').val(result.b_unit);
    $('#b_material_quantity').val(result.b_material_quantity);

  }).fail(function(xhr, status, err){
    console.log("bomList err:", err);
  });
}
```

콘솔창에 출력되는 bomList

```
bomList: ▼ (5) [{...}, {...}, {...}, {...}, {...}]
  ▶ 0: {rn: 1, b_material_id: '20000001', b_item_id: 10000001, m_name: '소나무', b_unit: 'm', ...}
  ▶ 1: {rn: 2, b_material_id: '20000002', b_item_id: 10000001, m_name: '소나무 다리', b_unit: 'ea', ...}
  ▶ 2: {rn: 3, b_material_id: '20000003', b_item_id: 10000001, m_name: '소나무 판', b_unit: 'ea', ...}
  ▶ 3: {rn: 4, b_material_id: '20000004', b_item_id: 10000001, m_name: '일반 나사', b_unit: 'ea', ...}
  ▶ 4: {rn: 5, b_material_id: '20000005', b_item_id: 10000001, m_name: '캠 나사', b_unit: 'ea', ...}
    length: 5
  ▶ [[Prototype]]: Array(0)
```

이를 활용해 전체 bom리스트를 가져오지 않고 Ajax와 자바스크립트를 사용해 웹 애플리케이션에서 AJAX 요청을 보내고, 서버로부터 응답을 받아와서 해당 데이터를 HTML 요소에 동적으로 추가하는 기능을 구현해보았다.

```
<script>

let bomListUL = $(".checkBomList");

function showBomList(i_id) { //i_id를 가져와서 화면에 뿌려주는 함수 선언 i_id=B_ITEM_ID

$.get("/bomList/" + i_id, function(result) { // 1. Bom 목록 rest ajax로 가져오기, ajax 함수 끝 성공시 처리

    console.log("showBomList:", result);

    var str = "";
    if (!result || result.length === 0) {
        bomListUL.html(""); // 제품에 등록된 bom이 없을 때 비우기
        return;
    }
    // 제품에 등록된 bom이 있을 경우 아래를 뿌려준다
    for (var i = 0; i < result.length; i++) {
        str += "<tr>";
        str += "<td>" + result[i].rn + "</td>";
        str += "<td>" + result[i].b_material_id + "</td>";
        str += "<td>" + result[i].m_name + "</td>";
        str += "<td>" + result[i].b_unit + "</td>";
        str += "<td>" + result[i].b_material_quantity + "</td>";
        str += "</tr>";
    }
    bomListUL.html(str); // 결과를 HTML에 삽입(html 렌더링한다)
}).fail(function(xhr, status, err) {
    console.error("showBomList error:", err);
});
}

// 특정 i_id에 대한 showBomList 호출, 가져와서 화면에 뿌려주는 함수 실행
showBomList(1);
```

위 코드는 JavaScript와 jQuery를 사용하여 웹 애플리케이션에서 AJAX 요청을 보내고, 서버로부터 받은 데이터를 화면에 표시하는 기능을 구현한 예시입니다.

JavaScript의 **showBomList(i_id)** 함수는 **bomService.getListBom()**을 호출하여 서버로부터 Bom 목록 데이터를 비동기적으로 요청합니다. 이 요청은 백엔드에서 처리되고, 요청이 성공하면 콜백 함수가 실행됩니다. 콜백 함수는 받은 데이터를 HTML 형식으로 구성하여 화면에 동적으로 표시합니다.

jQuery의 **\$.get** 메서드는 AJAX 요청을 보내고, 요청이 완료되면 서버에서 받은 데이터를 처리하는 역할을 합니다. 여기서는 서버에서 반환된 Bom 목록 데이터를 반복문을 통해 HTML 문자열로 생성하고, 이를 특정 HTML 요소 **<tbody>**에 삽입하여 화면에 표시합니다.

이렇게 구현된 코드는 웹 애플리케이션에서 동적으로 데이터를 가져와서 화면에 업데이트하는 일반적인 방식을 보여줍니다. JavaScript와 jQuery를 사용하여 AJAX 요청과 데이터 처리를 효과적으로 수행하는 예시입니다.

3. bom.jsp에서 제품번호를 클릭해 자재번호 수량 수정하기

자재번호의 수량을 수정하면 BOM관리의 모든 자재번호가 수정된다
즉 1개값이 아닌 여러개 값을 동시에 수정한다

이유: 수정시 자재번호의 값만 넘겨주었기 때문에 같은 자재번호가 있으면 모두 수정된다

Mapper.xml에서 SQL 쿼리문 수정

```
<update id="update">
    UPDATE BOM
    SET B_MATERIAL_QUANTITY = #{b_material_quantity}
    WHERE B_MATERIAL_ID = #{b_material_id}
    AND B_ITEM_ID = #{b_item_id}
</update>
```

ShowBomList 제이쿼리에서 b_item_id를 가지고오기 위해 data-itemid를 추가해준다

```
// 제품에 등록된 bom이 있을 경우 아래를 불러준다
for (var i = 0; i < result.length; i++) {
    str += "<tr>";
    str += "    <td>" + result[i].rn + "</td>";
    /* str += "    <td id='bomid-" + result[i].b_item_id + "' style='display: none; data-itemid='" + result[i].b_item_id + "'>" + result[i].b_item_id + "</td>";
    str += "    <td>" + result[i].m_name + "</td>";
    str += "    <td>" + result[i].b_unit + "</td>";
    str += "    <td id='mQuantity-" + result[i].b_material_id + "'>" + result[i].b_material_quantity + "</td>";
    str += "    <td id='mQuantity-backup-" + result[i].b_material_id + "' style='display: none;'>" + result[i].b_material_quantity + "</td>";
    str += "    <td>";
    str += "        <div class='modifyBom' id='modifyBom-" + result[i].b_material_id + "' data-bmid='" + result[i].b_material_id + "' data-itemid='" + result[i].b_item_id + "'>";
    str += "    </td>";
    str += "</tr>";
}

bomListUL.html(str); // 결과를 HTML에 삽입(html 렌더링한다)
```

위의 div class=modifyBom 수정버튼 클릭시 처리, \$('#mQuantity-' + bmid)를 replaceWith,
input을 넣어준다
후의 bomUpdate 안의 값을 두 값을 넣어준다 (bmid 자재번호, itemId 제품번호)

```
// 댓글 수정 버튼 처리
$('#modifyBom').on('click', function(e) {
    var bmid = $(this).attr('data-bmid'); // BOM 자재ID 가져오기
    var itemId = $(this).attr('data-itemid'); // BOM 제품ID 가져오기

    const mQuantity = $('#mQuantity-' + bmid).text(); // 실제 내용 가져오기
    console.log('modifyBom', mQuantity);
    tempQuantity = mQuantity;

    $('#mQuantity-' + bmid).replaceWith("<input type='text' id='mQuantity-" + bmid + "' value='" + (mQuantity || $('#mQuantity-' + bmid).val()) + "' />");
    /* $('#mQuantity-' + bmid).replaceWith("<td id='mQuantity-" + bmid + "' value='" + (mQuantity || $('#mQuantity-' + bmid).val()) + "'></td>"); */

    //CSS 적용
    $('#mQuantity-' + bmid).css({
        'width': '32px',
        'position': 'relative',
        'bottom': '-47px',
        'left': '35px'
    });

    bomUpdate(bmid, itemId); // BOM 수정 처리
});
}).fail(function(xhr, status, err) {
    console.error("showBomList error:", err);
});
});
```

```

@RequestMapping(method = { RequestMethod.POST }
    ,value = "/bom/{b_material_id}/{b_item_id}"
)
public ResponseEntity<String> modify(
    @RequestBody BomVO bomVO,
    @PathVariable("b_item_id") int b_item_id,
    @PathVariable("b_material_id") int b_material_id
){
    bomVO.setB_material_id(b_material_id);
    bomVO.setB_item_id(b_item_id);

    log.info("b_material_id: " + b_material_id);
    log.info("modify: " + bomVO);
    boolean isSuccessModify = bomService.modify(bomVO) == 1;

    return isSuccessModify ? new ResponseEntity<>{ body: "success", HttpStatus.OK)
        : new ResponseEntity<>(HttpStatus.INTERNAL_SERVER_ERROR);
}

```

Ajax로 수정처리를 해준다

```

//수정 버튼 실제 처리
function bomUpdate(bomMaterialId, bItemId) {

    console.log('bItemId:', bItemId);

    const bom = {
        b_item_id: bItemId,
        b_material_id: bomMaterialId,
        b_material_quantity: $('#mQuantity-' + bomMaterialId).val()
    };

    $.ajax({
        // request처리
        type : 'post',
        url : "/bom/" + bomMaterialId + "/" + bItemId, // form의 method속성 값 // form의 action값
        //url : "/bom/" + bomMaterialId, // form의 action값
        data : JSON.stringify(bom), // json으로 string처리하면서 파라미터 전달
        contentType : "application/json; charset=utf-8", // content-type지정
        // response처리
        success : function(result, status, xhr) { // call 성공시 오는 처리되는 함수
            $('#mQuantity-backup-' + bomMaterialId).text($('#mQuantity-' + bomMaterialId).val());
            console.log('zzzzzzzzzz', $('#mQuantity-' + bomMaterialId).val());
            console.log('tempQuan22', tempQuantity);

            // 그림 원래대로
            /* console.log('aaaa:', $('#mQuantity-' + bomMaterialId).prop('tagName')); */
            if (tempQuantity != $('#mQuantity-' + bomMaterialId).val()) {
                $('#mQuantity-' + bomMaterialId).replaceWith("<td id='mQuantity-" + bomMaterialId + "'>" +
                /* tempQuantity = $('#mQuantity-' + bomMaterialId).val(); */
            }
            /* console.log('bbbb:', $('#mQuantity-' + bomMaterialId).prop('tagName')); */
        },
        error : function(xhr, status, er) { // call 실패시 오는 처리되는 함수
            if (er) {
                console.log('error:', er);
            }
        }
    });
}

```

해결방법:

1. 함수밖에서 → `let tempQuantity = 0;` 선언
2. `$('.modifyBom').on('click', function(e) {` 여기 안에서
`const mQuantity = $('#mQuantity-' + bmid).text();` // 실제 내용 가져오와서
`tempQuantity = mQuantity;` 선언
3. 실제 수정을 처리하는 함수
`function bomUpdate(bomMaterialId, bltemId)` 안에서 **success** 한다면

`$('#mQuantity-backup-' + bomMaterialId).text($('#mQuantity-' + bomMaterialId).val());` 하고

`if (tempQuantity != $('#mQuantity-' + bomMaterialId).val()) {`

`$('#mQuantity-' + bomMaterialId).replaceWith("<td id='mQuantity-' + bomMaterialId + '>" + $('#mQuantity-' + bomMaterialId).val() + "</td>");`

`}`

이전에 넣었던 값 `tempQuantity`이 바뀐 값과 다르다면 다시 `tb`로 바꿔준다.

여기서 `tempQuantity`의 값은 다시 초기화되어 값은 `null`로 자동변경되어서 값이 이전과 같더라도 `tb`로 화면이 바뀌어진다.

4. 중복된 자재번호가 화면에 출력된다

자재 번호	20000001
	--choose--
자재 수량	20000001
1	20000002
	20000003
	20000004
	20000005
	20000001
	20000002
	20000003
	20000004
	20000005
	20000006

오류: 제품과 자재가 디비에 들어가 있는데도 등록이 가능하다

`ALTER TABLE FINAL2404_PROJECT2.BOM`

`ADD CONSTRAINT BOM_UNIQUE UNIQUE (B_ITEM_ID, B_MATERIAL_ID) ENABLE;`

디비에 두 값을 `UNIQUE`로 제약조건을 걸어주면 된다

삭제는 `unique`한 `id`가 없기 때문에 제품번호랑 자재번호를 둘이 넣어주고 그것이 맞을때 삭제된다

5. bomInsert.jsp BOM추가시 테이블 생성

```
/* var str = "";

str += "<tr>";
str += "    <td>";
str += "        <input class='form-check-input' type='checkbox' />";
str += "    </td>";
str += "    <td>" + itemId + "</td>";
str += "    <td>" + mtId + "</td>";
str += "    <td>" + mtUnit + "</td>";
str += "    <td>" + mtQuantity + "</td>";
str += "</tr>";
```

addBomListUL.html(str); // 결과를 HTML에 삽입(html 렌더링한다)

위를 사용하지 않는 이유:

1. 반복적인 문자열 연산이 필요하고 성능에 부담을 줄 수 있고,
2. 클릭할 때마다 기존 내용을 대체(**replace**)하는 것이 아니라, 기존 내용에 새로운 요소를 추가하려면 다른 방법이 필요 */

```
$.ajax({
    // request처리
    type : 'post',
    url : '/bom/' + itemId + "/" + mtId + "/" + mtUnit + "/" + mtQuantity,
    data : JSON.stringify({ itemId: itemId, mtId: mtId, mtUnit: mtUnit, mtQuantity: mtQuantity }),
    contentType : "application/json; charset=utf-8",
    // response처리
    success : function(result, status, xhr) {
        // call 성공시 오는 처리되는 함수
        // 새로운 행(tr)을 생성하여 HTML 문자열로 저장
        var newRow =
            "<tr>" +
            /* "<td><input class='form-check-input chk' type='checkbox' /></td>" + */
            "<td class='itemId'>" + itemId + "</td>" +
            "<td class='mtId'>" + mtId + "</td>" +
            "<td class='mtUnit'>" + mtUnit + "</td>" +
            "<td class='mtQuantity'>" + mtQuantity + "</td>" +
            "<td class='modifyBomInserted'>수정</td>" +
            "</tr>";

        // 생성한 행을 테이블에 추가
        $('#addBomListUL').append(newRow);
        alert('BOM 등록 성공');
    },
});
```

6. bomInsert.jsp 등록된 **bom**을 수정처리시 **mapper.xml**과 **controller**를 아래처럼 할 시에 단위 또는 수량은 수정되지만 모든 값을 다 넣으면 수정처리가 되지 않는다

```
<update id="updateAll">
    UPDATE BOM
    SET B_ITEM_ID = #{b_item_id},
        B_MATERIAL_ID = #{b_material_id},
        B_UNIT = #{b_unit},
        B_MATERIAL_QUANTITY = #{b_material_quantity}
    WHERE B_MATERIAL_ID = #{b_material_id}
    AND B_ITEM_ID = #{b_item_id}
</update>
```

```
@RequestMapping(method = { RequestMethod.POST }
    , value =
"/bom/modifyAll/{b_item_id}/{b_material_id}/{b_unit}/{b_material_quantity}"
)
public ResponseEntity<String> modifyAll(
    @RequestBody BomVO bomVO,
    @PathVariable("b_item_id") int b_item_id,
    @PathVariable("b_material_id") int b_material_id,
    @PathVariable("b_unit") String b_unit,
    @PathVariable("b_material_quantity") int b_material_quantity
){
    bomVO.setB_material_id(b_item_id);
    bomVO.setB_item_id(b_material_id);
    bomVO.setB_unit(b_unit);
    bomVO.setB_material_quantity(b_material_quantity);

    log.info("modify: " + bomVO);
    boolean isSuccessModify = bomService.modifyAll(bomVO) == 1;

    return isSuccessModify ? new ResponseEntity<>("success", HttpStatus.OK)
        : new ResponseEntity<>(HttpStatus.INTERNAL_SERVER_ERROR);
}
```

이유: **B_ITEM_ID**와 **B_MATERIAL_ID** 는 주요키 **FK**로 바뀌줄 수 가 없다.

해결: 컨트롤러에서 주요키만 받을 수 있도록 설정하고 수정은 단위와 수량만 수정할 수 있도록 설정해준다

7. bomInsert.jsp 등록된 bom을 수정처리시 수정 후 화면처리 안 됨

제품번호와 자재번호를 제외한 두 값을 input으로 바꿔준다. 이 때 **JQuery로 class명을 가져오고** if 문을 넣어 두 값을 제외한 나머지를 적용한다.

```
// '수정' 버튼 클릭 이벤트 핸들러 등록
$('.addBomListUL').on('click', '.modifyBomInserted', function() {
    var $row = $(this).closest('tr');

    // 각 셀의 값을 입력 필드로 변경
    /* $row.find('td').each(function() {
        var currentValue = $(this).text();
        $(this).html('<input type="text" class="form-control" value="' + currentValue + '">');
    }); */
    $row.find('td').each(function() {
        var currentValue = $(this).text();
        var className = $(this).attr("class");

        if (!(className == 'itemId' || className == 'mtId')) {
            $(this).html('<input type="text" class="form-control" value="' + currentValue + '">');
        }
    });
});
```

선언된 \$row 값을 다시 아래 한번 더 선언해준다. 이때 \$row2로 선언해주고 변경된 input 값을 다시 td로 바꾸어준다.

수정된 값을 다시 선언

이때 콘솔을 이용해서 \$row와 \$row2 그리고 \$this 의 값을 확인해 볼 수 있다.

```
// '수정' 버튼을 '완료' 버튼으로 변경
$(this).text('완료').removeClass('modifyBomInserted').addClass('completeBomEdit');

// '완료' 버튼(수정 완료 버튼) 클릭 이벤트 핸들러 등록
$(this).on('click', function() {
    var $row2 = $(this).closest('tr');

    // 각 셀의 입력 필드 값을 가져와서 해당 셀의 내용으로 설정
    $row2.find('input').each(function() {
        var newValue = $(this).val();
        $(this).closest('td').text(newValue);
    });

    // '완료' 버튼을 '수정' 버튼으로 변경
    $(this).text('수정').removeClass('completeBomEdit').addClass('modifyBomInserted');
    /* $row.find('.completeBomEdit').text('수정').removeClass('completeBomEdit').add

    // 수정된 데이터를 서버로 전송하여 처리 (실제 AJAX 요청)
    var itemId = $row2.find('.itemId').text();
    var mtId = $row2.find('.mtId').text();
    var mtUnit = $row2.find('.mtUnit').text();
    var mtQuantity = $row2.find('.mtQuantity').text();

    var dataToUpdate = {
        b_item_id: itemId,
        b_material_id: mtId,
        b_unit: mtUnit,
        b_material_quantity: mtQuantity
    };

    console.log(dataToUpdate);

    console.log('$row', $row);
    console.log('$row2', $row2);
    console.log('$this', $(this));
```

```

$row ▼ ce.fn.init i
  ▶ 0: tr
    length: 1
  ▶ prevObject: ce.fn.init {0: td.completebomedit, length: 1}
  ▶ [[Prototype]]: Object

$row2 ▼ ce.fn.init i
  ▶ 0: tr
    length: 1
  ▶ prevObject: ce.fn.init {0: td.completebomedit, length: 1}
  ▶ [[Prototype]]: Object

$(this) ▼ ce.fn.init i
  ▶ 0: td.completebomedit
    length: 1
  ▶ [[Prototype]]: Object

```

Ajax 실행을 하고 success 되었을 때 아래의 function을 넣어준다 여기서 **\$(this)** 는 td다 또 다시 JQuery로 class 명을 가져올 수 있도록 **className**으로 선언해주고 아래 if 조건문을 사용해 각 class 값에 맞는 데이터, 전에 선언 된 값을 넣어줄 수 있다.

```

// AJAX 요청을 통해 서버에 데이터 전송
$.ajax({
  type: 'POST',
  url: '/bom/modifyAll/' + itemId + '/' + mtId, // 수정 처리를 위
  contentType: 'application/json; charset=utf-8',
  data: JSON.stringify(dataToUpdate),
  success: function(response) {
    alert('BOM 수정이 완료되었습니다.');
```

```

    $row.find('td').each(function() {
      var className = $(this).attr("class");

      if (className == 'mtUnit') {
        $(this).html(mtUnit);
      }
      if (className == 'mtQuantity') {
        $(this).html(mtQuantity);
      }
    });
  },
  error: function(xhr, status, er) {
    alert('BOM 수정 중 오류가 발생했습니다. 다시 시도해 주세요.');
```

```

    console.error('error:', er);
  },
});

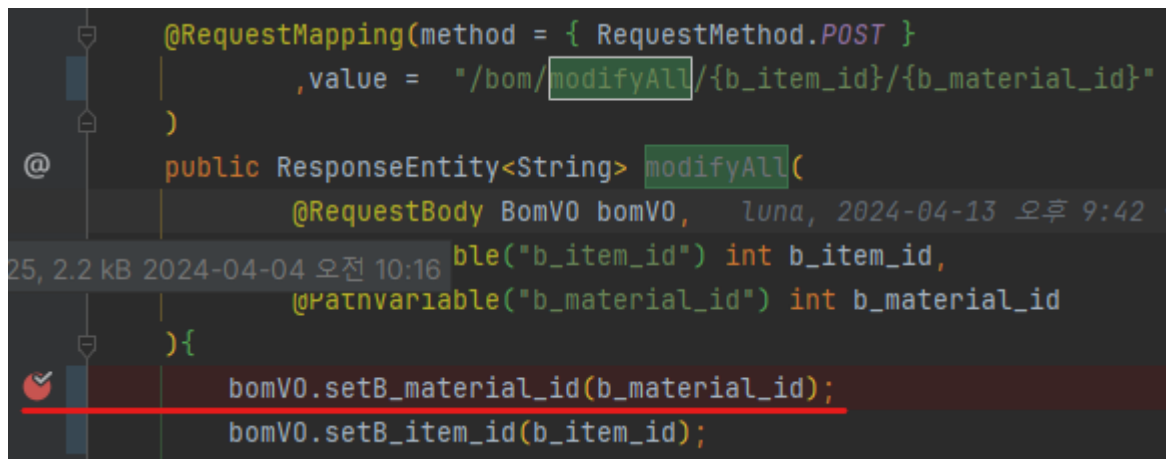
```

8. Ajax 등록 또는 수정처리시 인텔리제이 콘솔에 3번이상 실행되는이유?

- DBConfig Bean에 저장된 아래 때문일 수도 있지만, 디버깅을 통해서 정말 한 번만 실행되는지 확인할 수 있다

```
hikariConfig.setDriverClassName("net.sf.log4jdbc.sql.jdbcapi.DriverSpy");
```

- Controller 에서 디버깅 걸어주기 > 수정처리하는 웹 실행 > F9번을 눌러서 다음으로 넘어간다. 이 때 실행이 한번만 되는 것을 확인 (콘솔창에 실행이 여러번되도 신경쓰지 않아도 된다..?(선생님왈))



9. worker table 내용이 콘솔에 빈값으로 나온다

이유: Mapper.xml에 컬럼명 as 이름1 DTO에는 변경된 이름1이 아니라 컬럼명을 그대로 사용했다

ex) Mapper.xml: select worker_name as name

DTO: private String worker_name

즉, xml에 선언한 컬럼이름과 DTO의 컬럼이름이 매칭이 되지 않아 값을 못찾아 콘솔에 나오지 않았다.

해결: DTO에 원하는 컬럼명을 선언하고, xml에 resultMap을 사용해 컬럼명을 잘 찾을 수 있도록 매칭시켰다.

10. fullcalendar 라이브러리를 사용하는데 조회된 **worker table**의 내용이 스케줄에 적용되지 않는다

fullcalendar.io/docs

1) Controller 의 오류

```
@ResponseBody
@GetMapping("/schedule")
public String getScheduleAll() {
    List<ScheduleVO> scheduleList = scheduleService.getScheduleAll();
    return scheduleList;
}
```

public String이 아닌 List로 선언해줘야 한다.

```
@ResponseBody
@GetMapping("/schedule")
public List<ScheduleVO> getScheduleAll() {
    List<ScheduleVO> scheduleList = scheduleService.getScheduleAll();
    return scheduleList;
}
```

String 반환하는 경우:

메소드는 `@ResponseBody`를 사용하지만 String을 직접 반환합니다. 여기서 `scheduleList`는 `List<ScheduleVO>` 타입인데, 이것을 String으로 반환하려고 하면 Spring이 `List<ScheduleVO>` 객체를 문자열 표현으로 변환하려고 합니다. 반환된 `List<ScheduleVO>`가 Spring에 의해 자동으로 직렬화되지 않습니다. 이 접근 방식은 복잡한 객체를 간단한 문자열로 직렬화할 때 의도치 않은 결과를 초래할 수 있습니다. 결과적으로 응답은 List 객체의 문자열 표현인 `[object Object]` 또는 유사한 형태가 될 수 있습니다.

List<ScheduleVO> 반환하는 경우:

`getScheduleAll()` 메소드는 `@ResponseBody`로 주석 처리되어 있어 반환된 객체(`List<ScheduleVO>`)가 HTTP 응답 본문으로 직렬화되어야 함을 나타냅니다. Spring은 `scheduleList`를 JSON(기본값) 또는 요청의 Accept 헤더에 따라 다른 형식으로 자동 직렬화합니다. 응답 콘텐츠 유형은 일반적으로 `application/json`이 됩니다. 실제 `List<ScheduleVO>` 객체를 반환하며, Spring이 자동으로 JSON(또는 다른 형식)으로 직렬화합니다.

ResponseEntity<String> 반환하는 경우:

`getScheduleAll()` 메소드는 `scheduleService.getScheduleAll()`을 호출하여 `List<ScheduleVO>`를 가져옵니다. 이 리스트를 JSON 형식의 문자열로 직렬화한 후 `ResponseEntity.ok(jsonScheduleList)`를 사용하여 HTTP 응답을 생성하여 반환합니다.

결론:

`ResponseEntity<String>` 과 `List<ScheduleVO>`이 두 메소드는 반환하는 방식이 다르다. `ResponseEntity<String>` 메소드는 명시적으로 `ResponseEntity`를 사용하여 HTTP 응답을 생성하고 커스터마이징할 수 있다. `List<ScheduleVO>` 메소드는 Spring이 기본적으로

제공하는 `@ResponseBody` 기능을 이용하여 간단히 `List<ScheduleVO>`를 반환하며, Spring이 자동으로 JSON 형식으로 직렬화하여 응답한다.

따라서 같은 동작을 하지만 반환하는 방식이 다르며, 사용 목적과 상황에 따라 적절한 방식을 선택해야 합니다.

`@ResponseBody`를 사용한 **String** 반환 형태의 메소드는 객체를 직렬화하여 문자열로 변환하려는 의도를 가지고 있지만, 실제로는 문제가 발생할 수 있다.

이 메소드는 기술적으로는 올바르지 않습니다. `List<ScheduleVO>`를 직접 **String**으로 반환하는 것은 오류를 유발할 수 있다. 객체를 문자열로 변환하기 위해서는 명시적인 직렬화 작업이 필요하다.

대부분의 경우, 객체를 **JSON** 등의 형식으로 변환하여 반환할 때는

`ResponseEntity<String>`이나 `List<ScheduleVO>`를 그대로 반환하는 방식을 사용하는 것이 적절하다. Spring이 자동으로 직렬화를 수행하며, `ResponseEntity`를 사용하면 HTTP 응답 코드와 헤더를 더욱 세밀하게 제어할 수 있다.

2) <script> 안의 내용을 화면처리하는 ajax의 오류

이유:

ajax 사용시 필요한 **contentType: "application/json; charset=utf-8"**을 넣어주지 않았다.
data 매개변수는 FullCalendar의 **events** 함수의 특별한 요청 매개변수가 아니므로 불필요.

해결:

contentType: "application/json; charset=utf-8"를 넣어주고 **events** 함수는 FullCalendar에서 제공하는 특수한 콜백 형식을 따라야 한다. FullCalendar에서 이 콜백을 호출할 때는 **fetchInfo, successCallback, failureCallback** 세 가지 매개변수를 전달.

함수내에서 정말 작동하는지 **console.log**를 사용해서 찍어본다.

//데이터 가져오는 이벤트

```
events: function(fetchInfo, successCallback, failureCallback) {
    console.log("들어왔다"); ✓

    // AJAX 요청
    $.ajax({
        type: "GET",
        url: "/schedule",
        contentType: "application/json; charset=utf-8", // content-type지정
        success: function(response) {
            console.log("success들어왔다"); ✓
            var events = [];
            if (response != null) {
                var result = response;
                // 서버에서 받은 데이터로 이벤트 배열 구성
                for (var i = 0; i < result.length; i++) {
                    events.push({
                        title: result[i].title,
                        start: result[i].start,
                        end: result[i].end
                    });
                }
                // 이벤트 배열을 FullCalendar에 전달하여 캘린더에 표시
                successCallback(events);
            } else {
                // 서버 응답 오류 처리
                failureCallback('Invalid response from server');
            }
        },
        error: function(xhr, status, error) {
            // AJAX 오류 처리
            console.error('Error:', error);
            failureCallback('Failed to fetch events');
        }
    });
}
```