

## 1. item\_info 설정하기

가. items (해당 메뉴에서 사용할 컬럼 설정)

1) required\_write, required\_modify

- 필수 값 지정(필수면 1, 아니면 0)

- 이 항목을 사용하기 위해서는 submitType이 write나 modify로 설정되어야 하고

<itui:submitValid items="{itemInfo.items}" itemOrder="{itemInfo[itemOrderName]}" />

필수로 추가해야 함(itemOrderName은 item\_info에서 설정한 '\*\_order')

2) object\_type

- 해당 항목의 input type

- 0:text, 1:textarea, 2:select, 22:multi select...(RBS\_OPTION\_INFO 참고)

3) format\_type

- 해당 항목의 입력 값의 종류

- 0:일반, 2:휴대폰, 3:이메일, 4:주소....(RBS\_OPTION\_INFO 참고)

4) master\_code

- select, multi select, radio, checkbox에 필요한 코드 항목을 지정

- 1차원 코드 : 관리자 페이지(RBISADM) > 기능관리 > 기능코드관리에서 등록

(RBS\_OPTION\_INFO에 등록됨)

- 다차원 코드 : 관리자 페이지(RBISADM) > 기능관리 > 기능분류관리에서 등록

(RBS\_ICLASS\_INFO)

5) textarea 설정

- content\_type:2, editor:1, editor\_utype:1으로 설정하면 스마트 에디터로 나옴

- content\_type:2 만 설정하면 일반 textarea로 나옴

6) 항목을 직접 만들어서 태그에서 사용가능

```
387      },
388      "trnReqm":{
389        "column_id":"TRNREQM"
390        , "item_name":"훈련대상요건"
391        , "required_write":0
392        , "required_modify":0
393        , "required_approve_write":1
394        , "required_approve_modify":1
395        , "format_type":0
396        , "object_type":1
397        , "content_type":2
398        , "write_type":0
399        , "modify_type":0
400        , "default_item":1
401        , "use_list": 1
402        , "use_view": 1
403        , "use_write": 1
404        , "use_modify": 1
405      }
```

- 태그에서 \${itemObj['required\_approve\_write']} 사용

7) 그 외 다른 항목을 사용을 안해봐서 모르겠음....ㅎ

나. \*\_order (화면에서 사용할 items 항목을 설정)

1) objectView, objectText, objectSelect...와 같은 태그를 사용할 때 사용

2) \*부분은 자바에서 설정하는 submitType과 이름이 동일해야 함

3) master\_code를 사용하는 항목을 지정해두면 컨트롤러에서 해당 코드 정보를 가져옴  
(HashMap<String, Object> optnHashMap)

4) 여기서 지정하지 않으면 태그에서 사용할 수 없음

5) master\_code가 있는 항목은 여기서 지정하면 편함 (view.jsp에서 알아서 코드에 맞

는 항목이름 가져옴, input.jsp에서 object\_type에 맞게 setting 해줌)

6) \*proc\_order는 insert/update할 때 사용

(1) 여기에 항목을 지정하면 자동으로 insert/update할 항목을 만들어줌(dataList)

(2) 자바에서 필수 값을 체크할 때 사용

다. list\_search (검색 항목 설정)

1) keyField

(1) 검색어를 직접 입력할 때 사용

(2) column\_id는 서브쿼리를 넣어도 문제없이 동작함

```
730@ , "list_search":{
731@   "keyField":{
732@     "query_name":"key"
733@     , "select_name":"keyField"
734@     , "items":{
735@       "bplNo":{
736@         "column_id":"BPL_NO"
737@         , "item_name":"사업장관리번호"
738@         , "use_full_value":1
739@       }
740@       , "bplNm":{
741@         "column_id":"(SELECT BPL_NM1 FROM HRD_BSK WHERE BPL_NO = B.BPL_NO)"
742@         , "item_name":"기업명"
743@       }
744@     } , "select_order":[
745@       "bplNo"
746@       , "bplNm"
747@     ]
748@   } , "key":{
749@     "items":{
750@       "confmStatus":{
751@         "column_id":"A.CONFM_STATUS"
752@         , "item_name":"상태"
753@         , "format_type":0
754@         , "object_type":2
755@         , "style_width":100
756@         , "master_code":"CONFM_STATUS"
757@       }
758@     }
759@   }
760@ }
```

(3) keyField items는 keyField select\_order에 추가해줘야 항목이 제대로 나옴

(4) use\_full\_value를 1로 설정하면 '='로 검색됨 (설정하지 않으면 like로 검색됨)

2) key

(1) 검색 항목을 선택할 때 사용

(2) object\_type:5(radio), 22(multi select), 2(select), 14(checkbox) 주로 사용

(3) master\_code를 지정하거나 연도 같은 경우 format\_type:19, start\_year:2020, end\_year:2024와 같이 설정하여 사용 가능

(4) keyField처럼 column\_id에 서브쿼리를 넣어도 동작하지만 서브쿼리에 ','가 들어가면 이상하게 인식함(이유를 모르겠음....)

3) searchForm\_order

(1) 화면에서 보여줄 항목을 설정

(2) keyField는 keyField를 추가하고 key는 items 항목을 추가한다.

```
802   }
803@   , "searchForm_order":[
804@     "keyField"
805@     , "confmStatus"
806@     , "cliInsttIdx"
807@   ] , "search_order":[
808@     "keyField"
809@     , "confmStatus"
810@     , "cliInsttIdx"
811@   ]
812@ }
```

#### 4) search\_order

- (1) 쿼리에서 검색할 항목을 설정
- (2) 보통 searchForm\_order와 동일함
- (3) 검색할 때 ' ' 가 들어가는 서브쿼리를 사용해야 한다면 여기서 빼고 자바에서 따로 처리해야 함

ex. 능력개발클리닉 기업의 연차로 검색하기

Controller

```
1371
1372 // 연차 검색
1373 String yearItemName = "is_year";
1374 String year = queryString.getString(yearItemName);
1375 if(!StringUtil.isEmpty(year)) {
1376     param.put("YEAR", year);
1377     model.addAttribute("isSearchList", new Boolean(true));
1378     model.addAttribute("year", year);
1379 }
1380
```

Xml

```
153 <select id="selectList" parameterType="java.util.HashMap" resultType="dataMap">
154 <include refid="rbs.commonDBMSMapper.PAGING_TOP"/>
155 SELECT
156 (SELECT
157     D.BPL_NM
158 FROM
159     HRD_BSK D
160 WHERE
161     D.BPL_NO = B.BPL_NO)
162 AS BPL_NM,
163 A.*,
164 B.BPL_NO AS BPL_NO,
165 FN_GET_DOCTOR_NAME(B.DOCTOR_IDX) AS DOCTOR_NAME,
166 TO_CHAR(SYSDATE, 'YYYY') - SUBSTR(B.VALID_START_DATE, 0, 4) + 1 AS YEARLY
167 FROM
168     ${targetTable} A, HRD_CLI B
169 <include refid="rbs.commonDBMSMapper.SELECT_DE_WHERE"/>
170 <if test="YEAR != null">
171     AND (TO_CHAR(SYSDATE, 'YYYY') - SUBSTR(B.VALID_START_DATE, 0, 4) + 1) = #{YEAR}
172 </if>
173 AND A.CONFIRM_STATUS NOT IN ('3', '20') / 임시재용(3), 퇴부(20)은 부서외가 또는 인공 목록에서 제외되며 /
174 AND A.CLI_IDX = B.CLI_IDX
175 AND B.ISDELETE = '0'
176 AND TO_CHAR(SYSDATE, 'YYYYMMDD') <![CDATA[<=]]> B.VALID_END_DATE
177 <if test="ISHEAD == null">
178     AND B.CLI_INSTIT_IDX = #{INSTT_IDX}
179 </if>
180 ORDER BY
181     A.CONFIRM_STATUS, A.REGI_DATE DESC
182 <include refid="rbs.commonDBMSMapper.COM_ORDER"/>
183 <include refid="rbs.commonDBMSMapper.PAGING_BOTTOM"/>
184 </select>
```

javascript (search\_order에서 지정하면 자동으로 선택한 값이 나오지만 search\_order에서 지정하지 않았을 때 선택한 값 나오게 설정)

```
35
36 var year = "${year}";
37
38 // 연차 검색 항목 채우기
39 for(var i = 1; i <= 3; i++){
40     $("#is_year").append($('<option>', {value: i, text: i + "년"}));
41     if(year != "" && i == year){
42         $("#is_year option[value='" + i + "']").prop("selected", true);
43     }
44 }
45
```

## 2. setting\_info 설정하기

```
1 {
2   "setting_info":{
3     "list_title":"성공 목록",
4     "input_title":"성공 등록",
5     "deletelist_title":"성공 삭제목록",
6     "dset_list_block":10,
7     "dset_list_count":20,
8     "use_paging":1,
9     "page_name":"page",
10    "idx_name":"planIdx",
11    "idx_column":"PLAN_IDX",
12    "design":"plan",
13    "use_file":1,
14    "use_new":1
15  }
16  "clinicDct_setting_info":{
17    "list_title":"농약개발관리비 계획 목록(관리자)",
18    "input_title":"농약개발관리비 계획 등록(관리자)",
19    "deletelist_title":"농약개발관리비 계획 삭제목록(관리자)",
20    "dset_input_npname": "IDX_VIEW",
21    "dset_list_block":10,
22    "dset_list_count":20,
23    "use_paging":1,
24    "page_name":"page",
25    "idx_name":"planIdx",
26    "idx_column":"PLAN_IDX",
27    "design":"plan",
28    "use_file":1,
29    "use_new":1
30  }
31 }
```

confModule을 사용할 때 추가

- 1) use\_paging : 페이징처리 사용 유무
- 2) dset\_list\_count : 한 페이지에 나올 목록 수
- 3) dset\_input\_npname : 등록/수정 후 이동할 화면
- 4) idx\_name, idx\_column : 연관된 테이블 pk로 하는 듯?
- 5) design : 디자인 폴더 이름

## 3. tag 사용하기

### 가. 종류

- 1) itemXX.tag : objectItemName와 objectXX이 같이 있는 태그
- 2) objectItemName.tag : item\_info items에서 설정한 item\_name을 보여주는 태그
- 3) objectView.tag : 입력한 값을 보여주는 태그, master\_code를 사용할 때 쓰면 편함
- 4) objectXX.tag(objectText, objectSelect,...) : 값을 입력할 때 사용하는 태그, 수정할 때 기존에 입력한 값을 setting해서 보여줌
- 5) searchFormItem.tag : 검색 창 태그
- 6) searchFormItemIn.tag : 검색 항목 설정 태그
- 7) pagination.tag : 페이징 태그

### 나. 그 외 설명

- 1) 데이터가 dt가 아니면 objDt에 setting 해주기(안하면 태그에서 dt를 기준으로 알아서 setting함)
- 2) objVal에 컬럼값을 지정해서 사용 가능(안하면 objDt와 동일한 기준으로 알아서 setting함)
- 3) 필요한 attribute가 있으면 태그에 추가해서 사용 가능
- 4) searchFormItem.tag속성 : formAction(검색이나 전체목록 클릭 시 이동할 페이지), itemListSearch(item\_info에서 가져올 검색정보 list\_search), searchHashMap(검색할 때 필요한 master\_code)
- 5) 검색 후 페이징 처리하기 : listUrl에 검색조건 붙이기

\* 검색 후 페이징 예시 코드

```
<c:if test='${!empty param.keyField || !empty param.is_prtbiz || !empty param.is_ncsLclasCd }'>
<c:set var="searchList"
value="&keyField=${param.keyField}&key=${param.key}&is_prtbiz=${param.is_prtbiz}&is_ncsLclasCd=${param.is_ncsLclasCd}" />
</c:if>
<pgui:pagination
listUrl='${DEVELOP_TRAINING_LIST_FORM_URL}&tpList=${tpList}${searchList}'
pgInfo='${paginationInfo}' imgPath='${imgPath}'
pageName='${elfn:getString(settingInfo.page_name, 'page')}' />
```

\* 그 외 기타 내용 정리

## 1. multi file

- <input type="file">을 복사해서 사용한다. label을 써야 한다면 label도 같이 복사해주기
- 한 화면에서 multi file을 두 개 이상 쓰는 경우 fileCheck.js에서 \$("#file") 이라고 되어 있는거 전부 다 itemId로 바꾸기 + itemId에 일련번호 추가하기
- 업로드 확장자를 추가로 제한하려면 fileCheck.js에서 fn\_checkMultiFile()에 확장자 확인하는 코드 추가하기(기본으로 제한되는 항목은 globals.properties 확인)

```
77 // 이미지&pdf 파일만 업로드해야 할 때 확장자 검사(varItemId는 해당 id에 맞게 추가)
78 if(varItemId == "photo" || varItemId == "bill"){
79     var varFileValue = $(theFileObj).val(); // 원본파일명
80     if(!fn_isValFill(varFileValue)) return true;
81
82     var varExtName = varFileValue.substring(varFileValue.lastIndexOf(".") + 1); // 원본파일확장자명
83     varExtName = varExtName.toLowerCase();
84
85     // 업로드 가능한 이미지 확장자 목록(후쪽에 필요하다면 추가나 삭제)
86     var imgExtArray = [".jpg", ".jpeg", ".png", ".gif", ".pdf", ".raw", ".svg"];
87     var varReVal;
88
89     for(var varIdx = 0; varIdx < imgExtArray.length; varIdx++){
90         if(varExtName == imgExtArray[varIdx]){
91             // 저장될 파일목록에 추가
92             fn_fileCheck(theFileObj);
93             return true;
94         }
95     }
96 }
97
98 alert(".jpg, .jpeg, .png, .gif, .pdf, .raw, .svg 형식의 파일만 업로드 가능합니다.");
99 return false;
100
101 }
102
```

## 2. multi select

- 다차원 코드의 모든 항목을 하나의 select box에 나열하여 숨김처리한다.
- 먼저 1차 코드만 복사하여 보여준다. (data-level=1)
- 1차 코드가 선택되면 선택된 값이 data-pcode값과 같고 data-level이 2인 코드만 복사하여 2차 select box에 보여준다.
- 이런 식으로 3차, 4차.....
- 마지막으로 선택된 값이 숨김처리된 select box에 값으로 선택되어 보내진다.