

릴레이션매니저

엔터티는 기본적으로 하나의 마스터 테이블을 생성합니다. 엔터티 데이터 항목(Column)은 마스터 테이블에 생성하여 데이터를 저장하고, 확장 시 Custom 테이블 기능을 이
용합니다. Custom 테이블은 마스터 테이블과 1:1의 관계가 됩니다.

다양한 업무 환경에서 데이터간 관계도 다양해 질수밖에 없습니다. 1:1뿐만 아니라 1:N, N:N등 다양한 관계구조를 가지게 됩니다. 릴레이션은 이러한 데이터간, 즉 엔터티간 1:
N의 관계를 정의하여 리스트 화면까지 표현하는 모듈입니다.

릴레이션을 사용하기 위해서 데이터 모델뿐만 아니라 화면에서 사용되는 아톰까지 기본적인 지식이 있어야합니다.

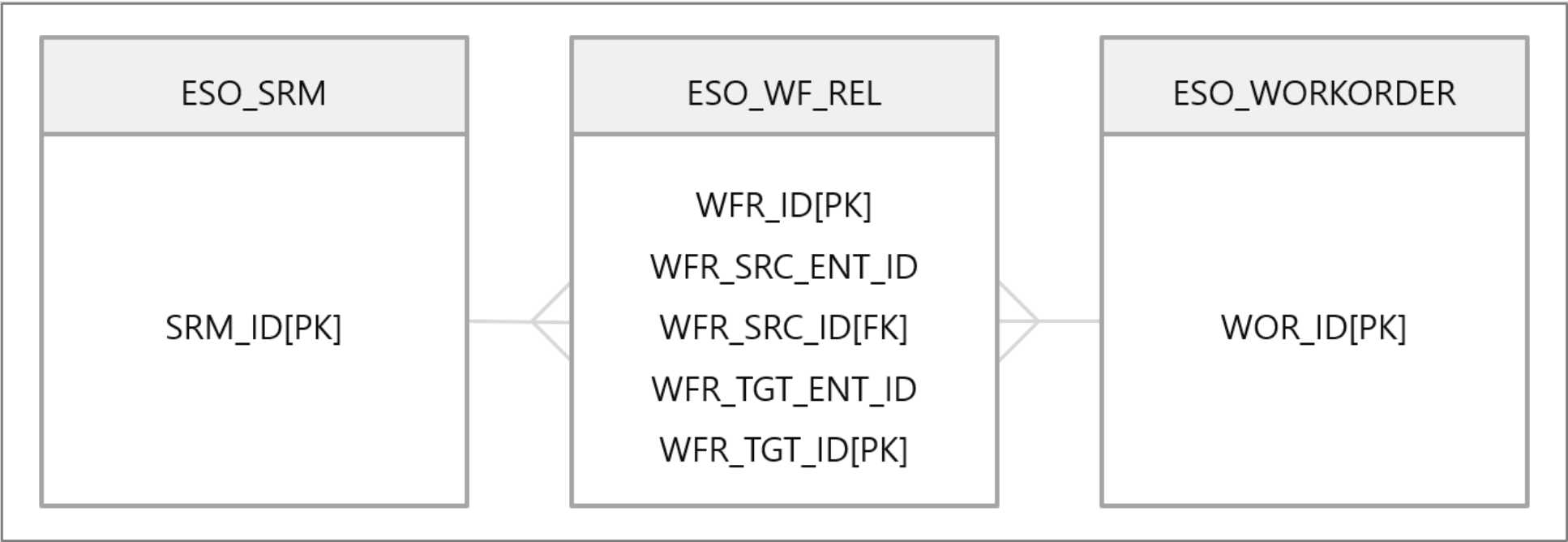
릴레이션 알아보기

데이터모델

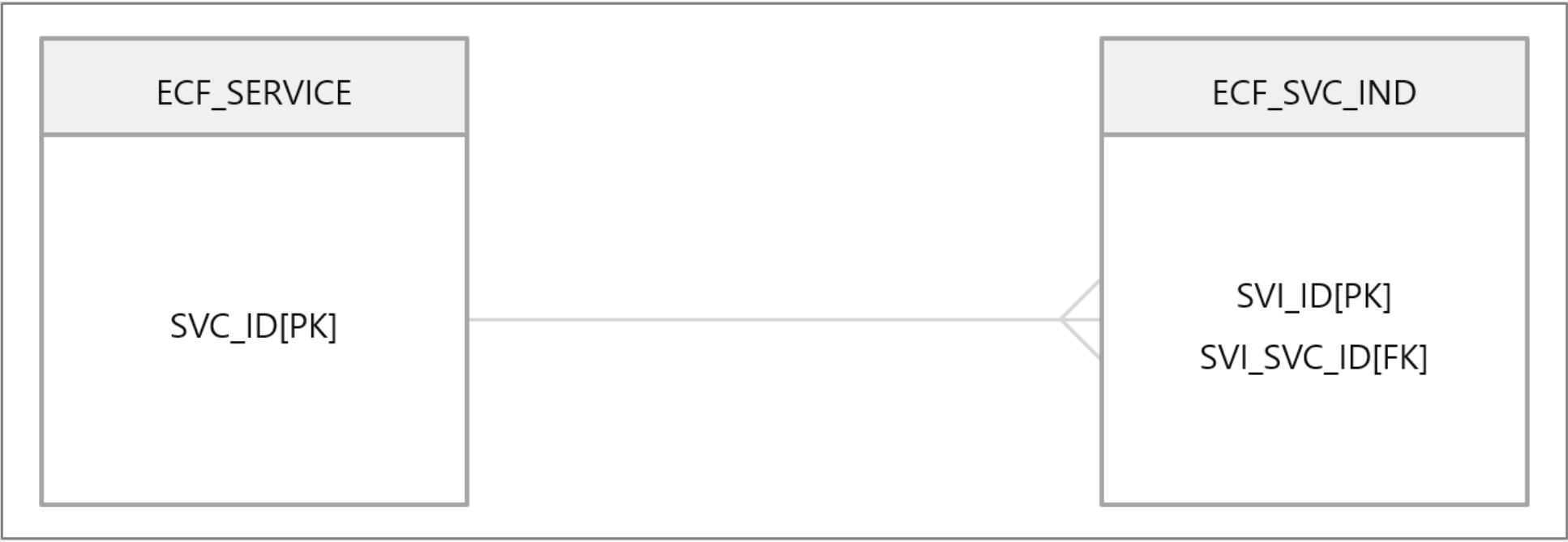
릴레이션은 두가지 방식의 데이터 모델로 구현됩니다. 하나는 엔터티간의 관계를 정의하는 릴레이션 테이블을 생성하여 관계를 매핑하는 방법이고, 또 하나는 엔터티간의 관계를 참조키(FK)를 이
용하여 관계를 매핑하고 부가정보를 관리 방법입니다.

일반적으로 폼화면에서 하나의 티켓에 여러 데이터를 입력하여 관리를 하기 때문에 두번째 방법이 많이 쓰이지만, 상황에 따라서 두개의 엔터티 정보를 매핑하는 경우가 발생하기 때문에 첫번째 방
법 또한 많이 사용됩니다.

N:N 데이터 구조 (연결 유형)



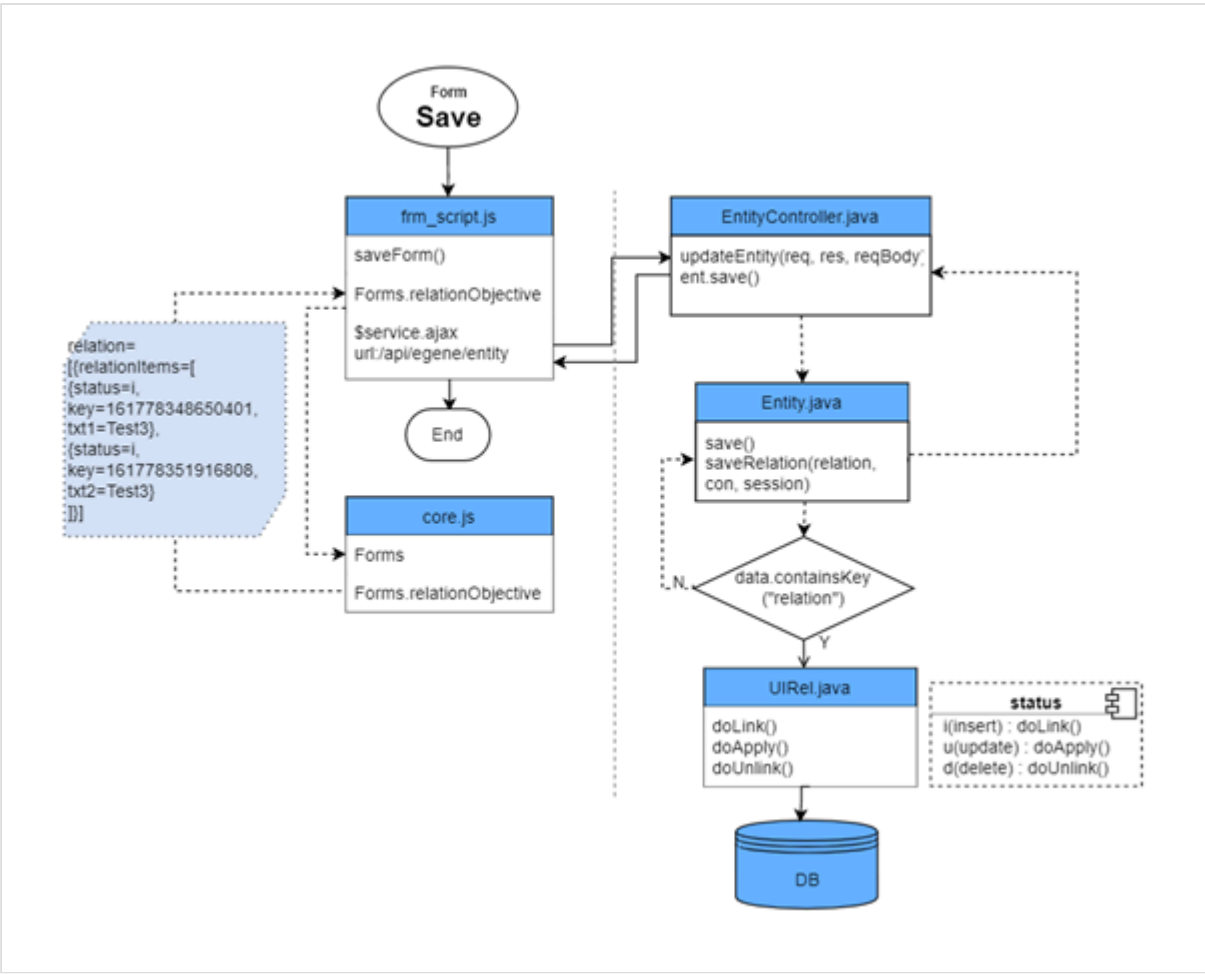
1:N 데이터 구조 (추가 유형)



저장방식
EGENE DOCUMENT

릴레이션은 폼 저장 요청 시 릴레이션 정보를 포함하여 하나의 트랜잭션으로 처리하여 저장하는 방식으로 구성되었습니다. 따라서 외부호출이나 메일 발송 등의 기능은 제어버튼에 설정하면 됩니다. 릴레이션 행삭제 시 최종 데이터 삭제 시점 역시 폼 저장 시점으로 변경되었습니다. 단, 릴레이션 추가 시 연결 리스트나 폼으로 등록하는 경우에는 기존 방식과 동일하게 등록과 동시에 저장됩니다.

status 값으로 처리 방식이 결정되며 'i'값은 추가, 'u'값은 저장, 'd' 삭제를 수행합니다.



API 저장방식

전달받은 릴레이션 데이터 값을 저장할 때 관계 엔터티의 API저장(entity save)을 수행합니다. 패키지 고유 저장 API이기 때문에 등록일, 수정일과 같은 컬럼을 설정하지 않아도 자동 업데이트 합니다.

SQL 저장방식

전달받은 릴레이션 데이터 값을 저장할 때 작성된 쿼리를 수행합니다. 쿼리는 Link SQL(추가), Unlink SQL(삭제), Apply SQL(수정) 3가지로 구분하여 작성합니다.

편집상태

릴레이션은 관계 데이터 수정을 목적으로 하는 아톰입니다. 데이터 편집에 대한 권한 제어가 중요하며, 권한에 따라서 상태에 따라서 데이터 편집에 대한 권한을 부여합니다.

릴레이션 아톰에 대한 권한 제어는 폼에서 설정하는 `readonly`(읽기)와 `editable` 두가지 상태가 있으며 편집 조건이 만족되어야 편집이 가능합니다.

설정에 따른 릴레이션 상태

Readonly	Editable	Column Editable	추가	연결	삭제	편집	표시

EGENE DOCUMENT	True	True	X	X	X	X	아름
		False	X	X	X	X	라벨
	False	True	X	X	X	X	라벨
		False	X	X	X	X	라벨
False	True	True	O	O	O	O	아름
		False	O	O	O	X	라벨
	False	True	O	O	O	X	라벨
		False	O	O	O	X	라벨

- Readonly : 폼 구성요소인 아름에 대한 상태 설정으로 폼 디자이너에서 읽기 상태가 체크되면 Readonly true 상태가 되어 편집 불가 상태가 된다.
- Editable : 릴레이션 아름 속성중의 하나로 릴레이션 아름 속성중 편집 상태가 체크되면 편집모드가 되어 설정된 아름이 표시된다.
- Column Editable : 기본값은 true이며 컬럼 속성중 editable = true|false 로 설정 가능하다.
- 추가 : 새로운 릴레이션 항목을 추가 한다. 릴레이션 리스트에 편집상태로 한줄 추가 하거나, 폼 화면을 팝업으로 표시하여 추가 할 수 있다.
- 연결 : 이미 존재하는 데이터를 리스트를 이용하여 표시하여 N개의 릴레이션을 추가할때 연결을 사용한다.
- 삭제 : 추가된 릴레이션을 체크박스를 통해서 선택하고 삭제한다.
- 편집 : 릴레이션 아름이 편집상태가 되면 설정된 아름을 표시 하여 데이터를 편집한다. 설정된 아름이 없다면 일반 텍스트를 입력하는 Input 박스가 표시된다.

릴레이션 생성하기

Relation은 관리자 메뉴 관계 관리 화면에서 관리됩니다. 관계관리 화면에서는 새로운 Relation을 추가하거나 등록된 Relation을 수정할 수 있습니다.

Relation 관리 속성

릴레이션 속성은 저장방식에 따라 설정해야 할 속성이 달라집니다.

저장방식	속성명	설명
공통	ID	릴레이션 고유 ID
공통	이름	릴레이션 명 (폼 저장시 릴레이션 라벨이 없는 경우 require시 표기될 명칭)
공통	Explorer 1	연결된 정보를 보여줄 릴레이션(explorer) 유형의 List
공통	Explorer 2	연결할 정보를 보여줄 연결 팝업(link) 유형의 List
공통	설정(Config)	릴레이션의 부가 설정 *상세 속성 내용 참조
공통	관계 엔터티	관계 정보가 저장될 엔터티 ID

EGENE DOCUMENT

SQL	Link SQL	연결 또는 신규 저장 시 실행할 DML을 작성
SQL	Link Argument	Link SQL의 Argument를 작성
SQL	Unlink SQL	삭제 시 실행할 DML을 작성
SQL	Unlink Argument	Unlink SQL의 Argument를 작성
SQL	Apply SQL	수정 저장 시 실행할 DML을 작성
SQL	Apply Argument	Apply SQL의 Argument를 작성
API	저장 항목 설정	DML 필드 맵핑 정보 설정

설정(Config) 상세 속성

릴레이션의 부가적인 설정을 합니다. 설정은 JSON Object 형태로 작성되며 릴레이션 매니저를 통해 부가 설정 단계에서 화면 UI로 구성할 수 있습니다.

속성명	설명
layout	연결할 List Popup의 Layout 설정, JSON Object 형태로 작성
layout.win_w	layout 하위 속성으로 윈도우 가로 길이 설정
layout.win_h	layout 하위 속성으로 윈도우 높이 설정
args	릴레이션의 조회 화면이나 DML 실행 시 전달할 파라미터 변수를 설정
autoAddRow	기본 true로 최초 화면에서 데이터가 없을 때 기본 row 보여주는 옵션, addType이 있으면 무조건 false
addType	릴레이션 추가를 폼 또는 별도 화면을 이용할 것인지 유형 설정 url form
form	addType이 form일경우 폼 아이디를 설정
create_uri	addType이 url일 경우 설정
link_uri	연결 시 릴레이션 기본 연결이 아닌 별도의 개발된 화면으로 연결 수행할 경우 URI를 설정
upload_uri	엑셀 업로드 사용할경우 업로드 uri 설정함
unlink_act	별도 개발된 삭제 프로그램 수행할 경우 URI 설정, ","를 구분자로 멀티 URI 설정 가능
link_act	별도 개발된 연결 프로그램 수행할 경우 URI 설정, ","를 구분자로 멀티 URI 설정 가능
apply_act	별도 개발된 업데이트 프로그램 수행할 경우 URI 설정, ","를 구분자로 멀티 URI 설정 가능

샘플

EGENE DOCUMENT

```
"layout":{
  "win_w":800,
  "win_h":580
},
"args":"arg1,arg2,arg3",
"addType":"form",
"form":"FRM00001",
"upload_uri":"/xefc/jsp/ui/popup_layout.jsp?uri=/xefc/jsp/loader/loader.jsp&ldr_id=LDR01"
}
```

Relation 관계 설정(API 저장방식)

API 저장방식 릴레이션의 경우 릴레이션 마법사를 통해 관계정보를 저장할 엔터티 생성부터 필드 설정, 리스트 화면까지 Step by Step으로 생성할 수 있으며 릴레이션 매니저를 통해 생성된 내용을 관리할 수 있습니다.

저장 항목 설정 속성

API 저장방식에서는 DML 필드 정보를 JSON Object 형태로 관리하며 릴레이션 매니저를 통해 항목설정 단계에서 화면 UI로 구성할 수 있습니다.
한번의 등록, 수정에 여러 필드가 대상이 되기 때문에 저장 항목 설정 속성은 fields 속성 하위에 필드 목록으로 관리합니다.

속성명	설명
fld_id	저장될 필드의 ID
type 1)	저장될 값이 들어있는 변수 유형
value	유형에 따라 저장될 값 또는 변수 값
label	저장될 필드의 이름
sequence	키 컬럼일 경우 시퀀스 설정
mod	수정 상황에서 해당 필드 저장 여부(true/false)
reg	등록 상황에서 해당 필드 저장 여부(true/false)
require	필수 필드 여부(true/false)

1) type
시스템값(D)/필드값(F)/예약어(S)/입력값(V)의 4가지 유형이 있으며 사용자는 시스템값을 제외한 3가지 유형을 선택할 수 있습니다.
시스템 값으로는 키 컬럼, 소스 키 컬럼, 타겟 키 컬럼 정보가 들어갑니다.

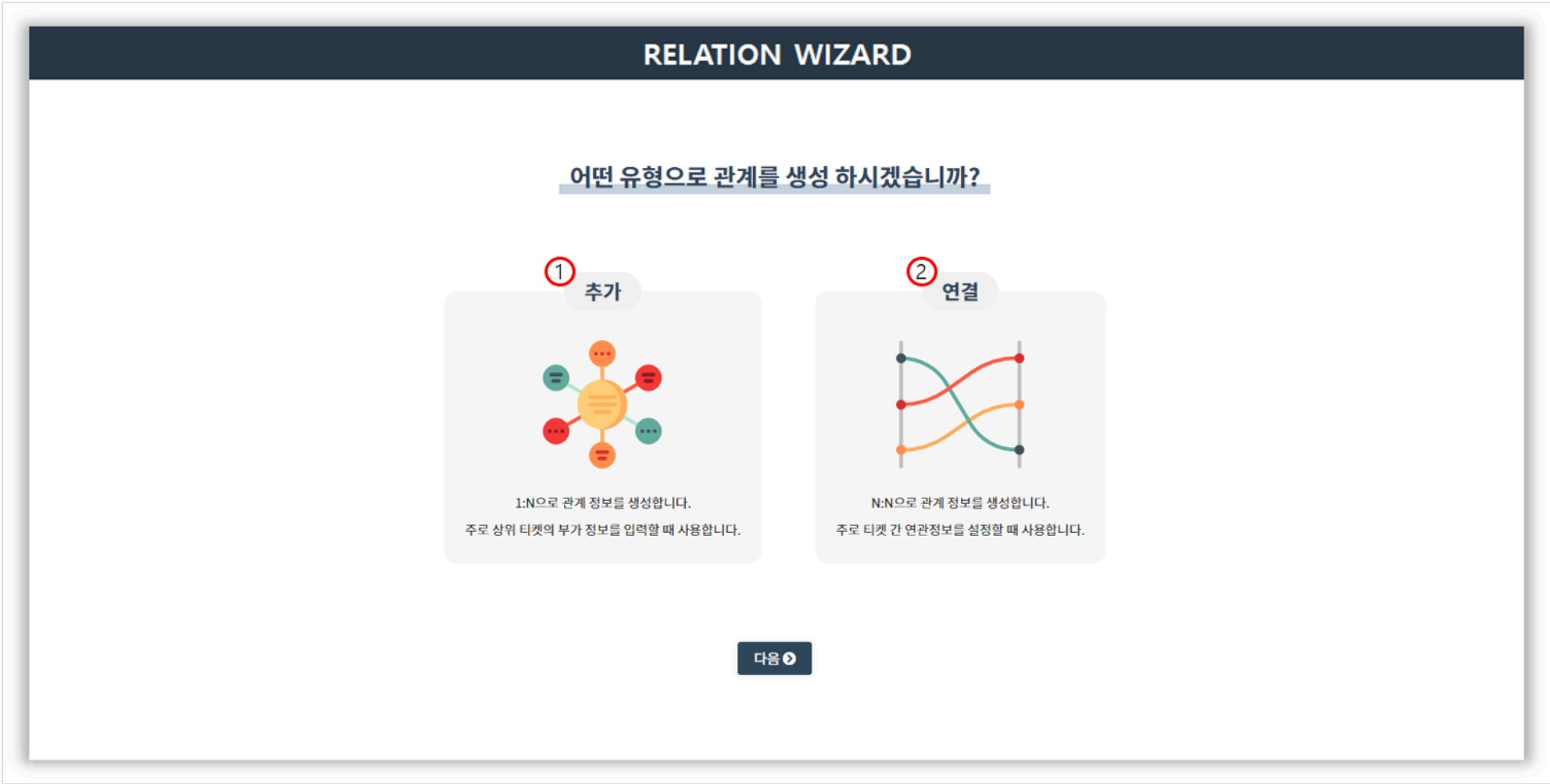
샘플

EGENE DOCUMENT

```
"fields": [
  {
    "fld_id": "ESO_TST/TST_ID",
    "type": "D",
    "label": "ID",
    "value": "key",
    "sequence": "ent_seq.TST",
    "mod": false,
    "reg": true,
    "require": true
  }
]
```

관계 유형 결정

릴레이션 유형 선택은 릴레이션 마법사를 실행하면 보여지는 최초 화면으로 데이터 모델을 결정합니다.



- 1. 추가
 - 1:N으로 관계 정보를 생성합니다.
 - 주로 상위의 부가 정보를 입력할 때 사용합니다.
- 2. 연결
 - N:N으로 관계 정보를 생성합니다.
 - 주로 티켓 간 연관정보를 설정할 때 사용합니다.

기본정보

1 ID

RLT00128

2 이름

릴레이션 001

3 사용여부

☒ 예
 ☐ 아니오

4 소스 엔터티

[SRM] SRM

5 타겟 엔터티

[042] Workflow

6 연관 엔터티

ID	이름	유형	마스터 테이블	Prefix
SRM	SRM	Workflow	ESO_SRM	srm

1. ID
2. 이름
3. 사용여부
4. 소스 엔터티

릴레이션 마법사 화면에서 보이며 해당 릴레이션을 사용할 엔터티를 뜻합니다.

폼 디자이너 또는 엔터티 관리에서 릴레이션을 생성할 경우 소스 엔터티는 해당 엔터티로 고정됩니다.

아틀

Custom.Relation

값 설정

ID

RLTWIZTEST001

+

⚙

Edit

☒

Control

☒

삭제

☒

엑셀 다운로드

☐

엑셀 업로드

☐

추가

☒

연결

☐

Args

릴레이션 선택

닫기

검색...

▶ 신규추가

[1]

[CH_RE_D43] 배포 요청 리스트

[CH_UT_D41] 단위테스트

[CH_WO_UTM01] AP변경 단위테스트 릴레이션 관리

[CHM_AGREEMENT] [변경] 완료계획일변경이력

[CHM_AGREEMENT_read] [변경] 완료계획일변경이력 READ

[CHM_CAB_INFRA] [인프라변경]CAB구성원

[CHM_CM_EDIT] 구성항목_속성변경

[CHM_INFRA_WOR] 인프라-작업지시

확인

5. 타겟 엔터티

릴레이션 마법사 화면에서 연결 유형일 때 보이며 연결 리스트에서 조회할 엔터티를 뜻합니다.

엔터티 정보에 따라 **연결 리스트(Explorer 2)**를 자동 생성합니다.

6. 연관 엔터티

릴레이션 매니저 화면에서 보이며 관계 엔터티와 연관된 엔터티의 목록입니다.

관계 엔터티 선택

관계정보가 쌓일 엔터티 정보를 설정합니다. 릴레이션 마법사에서는 관계 엔터티를 신규로 생성할 수 있으며 기존 엔터티 중 선택할 수 있습니다.

1. 관계 엔터티 신규 생성

테이블

관계 엔터티 및 테이블을 신규로 생성 하시겠습니까?

☒ 예
 ☐ 아니오

* 엔터티 ID

REL0043

* 이름

신규 엔터티 이름을 입력하세요

* 마스터 테이블

ESO_REL0043

* Prefix

rel

신규로 생성할 관계 엔터티 정보를 입력합니다. ID, 마스터 테이블, Prefix는 규칙에 맞춰 자동 생성 (REL0001|ESO_REL0001|rel) 되며 식별할 수 있는 이름을 입력합니다.

릴레이션에서 엔터티를 생성할 경우 엔터티 유형은 "Relation"으로 세팅됩니다.

테이블

관계 엔터티 및 테이블을 신규로 생성 하시겠습니까?

☐ 예 ☒ 아니오

관계엔터티

검색 할 엔터티를 입력해주세요.

[SMS] SMS

[SOL] SOL

[SQS] [실문조사] 질문 그룹 관리

[SQT] [실문조사] 질문 Relation 그룹

[SRA] SRA

[SRM] SRM

[SRT] 릴레이션 TEST

[STEGW01] STEGW01

ID	SRA
이름	SRA
마스터 테이블	ESO_SRA
Prefix	sra

전체 엔터티 목록 중에 관계 엔터티를 검색하여 선택합니다.

항목설정

DML 설정을 엔터티 필드 기준으로 Drag&Drop을 통해 쉽게 설정하며 릴레이션 마법사의 경우 **DML 설정 정보에 따라 조회 리스트(Explorer 1)**을 자동 생성합니다.

항목

필드

검색 할 필드를 입력해주세요.

tst_reg_dpt_id (등록자부서)

tst_mod_dttm (수정일시)

tst_mod_emp_id (수정자)

tst_order (순서)

tst_used (사용여부)

tst_src_id (소스 ID)

tst_tgt_id (타겟 ID)

tst_height (신장)

tst_test01 (test)

tst_test02 (test2)

tst_test03 (test03)

tst_test04 (test04)

tst_test05 (test05)

키 컬럼

tst_id (ID)

소스 키 컬럼

tst_src_id (소스 ID)

타겟 키 컬럼

tst_tgt_id (타겟 ID)

DML 설정

항목	컬럼	변수	필수여부	등록	수정
	tst_name (이름)	<div>필드값</div> tst_name (이름)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	tst_descr (설명)	<div>필드값</div> tst_descr (설명)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
	tst_test01 (test)	<div>예약어</div> 현재 일시	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	tst_test02 (test2)	<div>입력값</div> mod	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

1. 추가: 엔터티 필드를 추가합니다.

EGENE DOCUMENT

항목

필드

추가

키 컬럼

DML 설정

검색 할 필드를 입력해주세요.

tst_reg_dpt_id (등록자부서)

tst_mod_dttm (수정일시)

tst_mod_emp_id (수정자)

tst_order (순서)

tst_used (사용여부)

tst_src_id (소스 ID)

tst_tgt_id (타겟 ID)

tst_height (신장)

tst_test01 (test)

tst_id (ID)

소스

필드 추가

* ID

TST_

* 이름

* 유형

문자열(100)

감사

☐

닫기

확인

- 2. 키 컬럼: 관계 정보의 키가 될 컬럼이 설정됩니다.
- 3. 소스 키 컬럼: 소스 엔터티의 키가 저장될 컬럼을 설정합니다.
- 4. 타겟 키 컬럼: 연결 유형일 때 보이며 타겟 엔터티의 키가 저장될 컬럼을 설정합니다.
- 5. 변수

컬럼에 저장될 변수를 설정할 수 있으며 필드값, 예약어, 입력값의 세가지 유형이 있습니다.

필드값: 화면에서 선택한 값을 의미하며 조회 리스트의 필드 목록 중 선택합니다.

필드값

Search...

tst_tgt_id (타겟 ID)

tst_name (이름)

tst_descr (설명)

hide_data (숨김테스트)

필드

항목	* ID	라벨 ID	* 이름	*아름	*아
✓ <input type="checkbox"/> <input type="checkbox"/> ✕	tst_tgt_id	tst_tgt_id	타겟 ID	String.Text	<input type="checkbox"/>
✓ <input type="checkbox"/> <input type="checkbox"/> ✕	tst_name	tst_name	이름	Code.Select	<input type="checkbox"/>
✓ <input type="checkbox"/> <input type="checkbox"/> ✕	tst_descr	tst_descr	설명	Code.Select	<input type="checkbox"/>
✓ <input type="checkbox"/> <input type="checkbox"/> ✕	hide_data	hide_data	숨김테스트	String.Text	<input type="checkbox"/>

예약어: 기본적으로 릴레이션 키, 현재일시, 현재일, 로그인 사용자, 로그인 사용자 부서, 로그인 사용자 회사 중 선택할 수 있으며 "설정 탭"의 파라미터로 설정한 값이 있을 경우 해당 값을 선택할 수 있습니다.

예약어

Search...

릴레이션 키

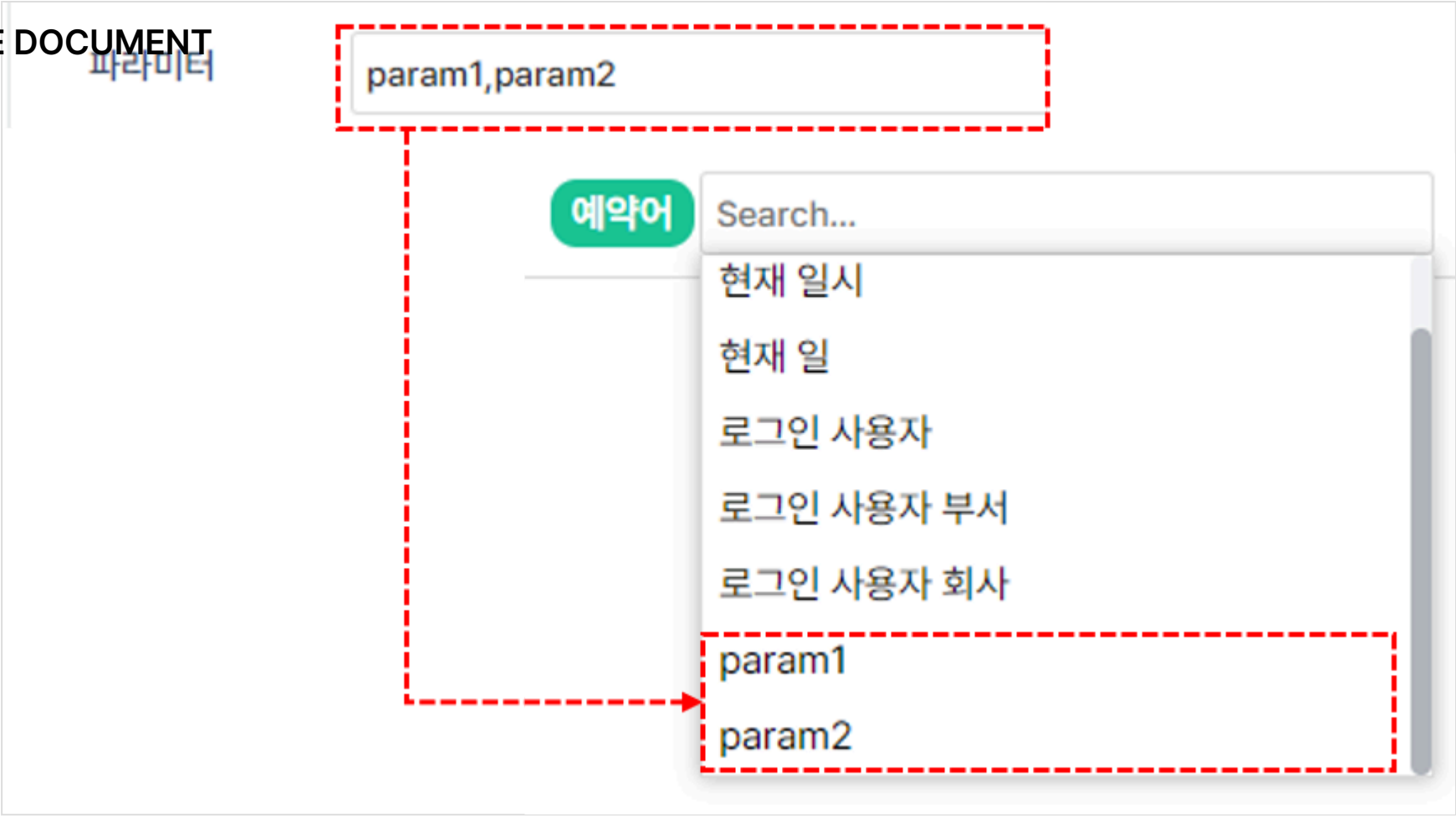
현재 일시

현재 일

로그인 사용자

로그인 사용자 부서

로그인 사용자 회사



- 입력값: 저장할 값을 직접 TEXT로 입력합니다.
- 필수여부: 릴레이션 마법사에서만 보여지며 조회리스트를 자동 구성할 때 필수필드 설정에 사용됩니다.
 - 등록/수정: 각각 체크되어 있는 상황에만 저장됩니다.

부가 설정

릴레이션 아토름을 위한 부가 정보를 설정합니다. 기존에 JSON으로관리하던 정보를 UI를 통해 설정할 수 있습니다.



- 조회 리스트: 릴레이션 매니저 화면에서 보여지며 폼 화면에 보여질 리스트를 의미합니다. 리스트 매니저로 상세설정을 수정할 수 있습니다.
- 연결 리스트: 릴레이션 매니저 화면에서 보여지며 연결 버튼을 눌렀을 때 팝업될 리스트를 의미합니다. 리스트 매니저로 상세설정을 수정할 수 있습니다.
- 추가유형: 데이터를 추가하는 형태를 결정합니다. 행, 폼 팝업, URI 팝업 중 선택 가능하며 기본값은 행 유형입니다.
 - 기본 행 사용: 예 일 경우 릴레이션 로드 시 화면에 빈 행을 보여줍니다. (저장된 데이터가 없을 때)
 - 폼: 폼 팝업 유형일 때 보여지며 팝업될 폼 ID를 입력합니다.
 - URI: URI 팝업 유형일 때 보여지며 팝업될 URI를 입력합니다.
- 팝업 넓이/팝업 높이: 연결 유형일 때 보여지며 팝업 넓이와 팝업 높이를 설정할 수 있습니다.
- 파라미터: 릴레이션 화면에서 사용할 파라미터를 설정합니다.
- 엑셀업로드: 데이터를 일괄로 업로드 하고자 할 때 엑셀 업로드 기능을 사용할 수 있습니다.

Relation 관계 설정(SQL 저장방식)

릴레이션 연결 대상 엔터티의 DML과 동적 Argument를 정의합니다.

DML 설정 관련 속성

- DML : 릴레이션 저장 시 실행되는 DML 구문
- Argument : DML 구문에 적용 될 Argument
- cols : 입력할 컬럼 정보
- require : 필수 필드, 미입력 시 데이터가 저장되지 않음

Argument의 cols에 사용할 변수 예약어 속성

EGENE DOCUMENT 속성명	설명
#key	Timestamp Type key
#rel_key	Form의 Source key
#ent_id	Entity ID
#cur_dttm	현재일시
#cur_dt	현재일
#cur_emp, #emp_id	로그인 사용자 ID
#cur_org, #org_id	로그인 사용자 회사 코드
#cur_dpt, #dpt_id	로그인 사용자 부서 코드
key	select row data의 key. #을 붙이지 않음
#type	form config의 args로 설정한 값을 사용 할 경우 반드시 '#'을 붙여서 사용 ex) form config { "args":"type" } 설정하면 릴레이션 DML argument 에는 { "cols":"#type,#rel_key" }

Link 설정(추가)

연결 또는 저장 시 실행되는 DML 구문입니다. 일반적으로 insert 문을 사용합니다.

Link DML

```
insert into eso_template(tmp_id, tmp_text_05, tmp_src_id)
values(?, ?, ?)
```

Link Argument

```
{
  "cols":"#key, tmp_text_05, #rel_key",
  "require":"#key,#rel_key"
}
```

Apply 설정(수정)

저장 시 실행되는 DML 구문입니다. 일반적으로 update 문을 사용합니다.

Apply DML

```
update eso_template
  set tmp_text_05= ?
where tmp_id = ?
```

Apply Argument

EGENE DOCUMENT

```
"cols": "tmp_text_05, key",
"require": "key"
}
```

Unlink 설정(삭제)

삭제 시 실행되는 DML 구문입니다. 일반적으로 delete 문을 사용합니다.

Unlink DML

```
delete from eso_template where tmp_id= ?
```

Unlink Argument

```
{
  "cols": "key",
  "require": "key"
}
```

Relation List (Explorer 1)

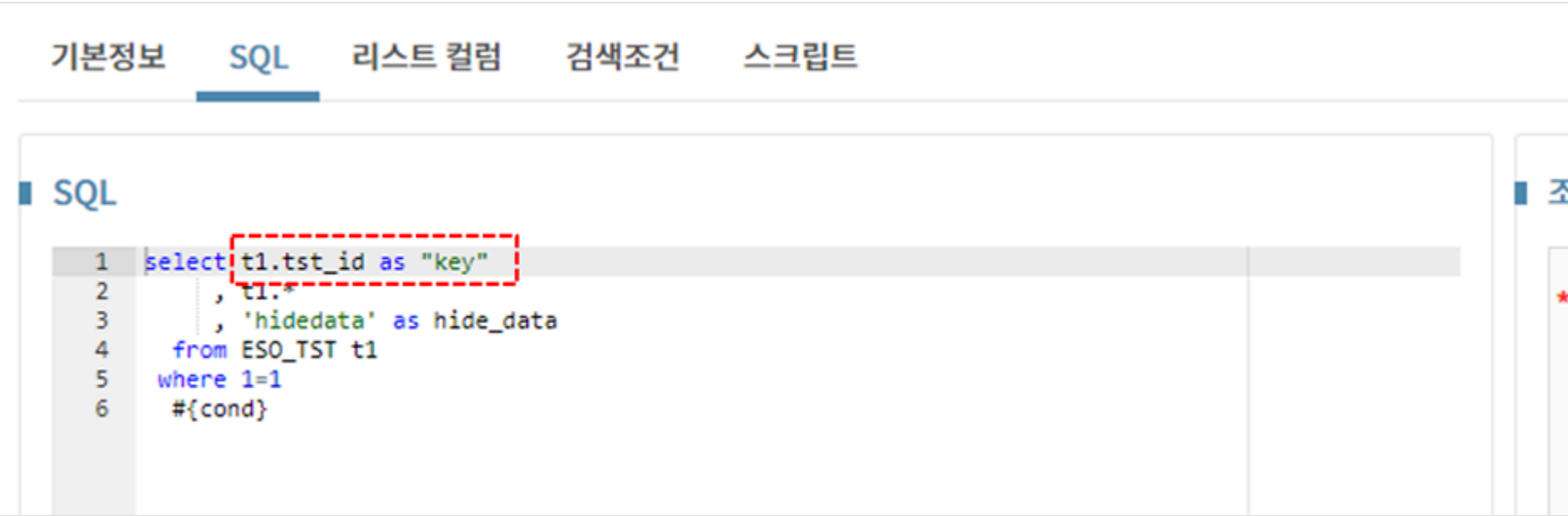
Relation List는 폼 화면에 표시되는 리스트 화면을 구성합니다.

리스트 매니저를 통해 구성하며 이 때 사용하는 유형은 “릴레이션 유형(Explorer)” 입니다.

자세한 리스트 화면 구성은 리스트 매니저 가이드의 리스트 생성하기 목차를 참고바라며, 리스트 유형은 “릴레이션 유형(Explorer)”입니다.

아래는 리스트를 구성할 때 반드시 설정해야 할 속성에 대한 설명입니다.

- 1. SQL에 Target Key로 저장할 컬럼을 “key” alias로 설정합니다.



1. 리스트 컬럼 설정

릴레이션 유형의 컬럼 속성은 아래와 같습니다. (* 필수 속성)

속성	설명
* ID	데이터를 보여 줄 컬럼. SQL 쿼리에 작성한 alias와 동일 해야 함.
라벨 ID	아utom에 저장되는 컬럼. 공백 시 ID 값으로 치환 됨.

이름	데이터의 헤더로 표시될 이름.
* 아톰	아톰 명. 아톰 선택 후 도움말 버튼 클릭 시 가이드 확인 가능.
* 아톰 설정	선택 한 아톰에 대한 추가 설정. 아톰 가이드로 설정 샘플 확인 가능.
* 길이	컬럼의 길이.
정렬	정렬 방식(왼쪽, 가운데, 오른쪽)
이벤트 설정	초기이벤트, 클릭이벤트, 변경이벤트 설정.
행	Custom.MultiSet 아톰에서 사용하는 속성. 폼 위치 정보이며 행 위치 값. 왼쪽부터 1로 시작하도록 설정.
열	Custom.MultiSet 아톰에서 사용하는 속성. 폼 위치 정보이며 열 위치 값. 위에서부터 1로 시작하도록 설정.
컬럼 영역	Custom.MultiSet 아톰에서 사용하는 속성. 폼 필드 컬럼 개수가 2개 이상일 경우 정의하며 필드가 차지하는 영역에 대한 설정. table 레이아웃에서 colspan 값에 해당하며 설정 값에 따라서 컬럼 영역이 할당.
유형	컬럼을 표시할 포맷 유형. 엑셀과 숨김을 지원한다. 엑셀: 화면에서는 해당 컬럼이 숨겨지나 엑셀 다운로드 시 포함된다. 숨김: 화면에서 해당 컬럼이 숨겨지고 데이터만 전달된다.
필수	컬럼 필수 여부
편집	컬럼 편집 여부

리스트 컬럼에서 설정한 내용은 화면에서 아톰으로 보이며 입력 받은 데이터가 ID를 변수명으로 하여 Link, Unlink, Apply가 수행될 때 함께 전달됩니다.

API 저장방식에서는 항목설정 따라 변수 값으로 사용할 수 있습니다.

SQL 저장방식에서는 Link, Unlink, Apply 설정에 따라 Arguments 설정의 cols 값으로 사용할 수 있습니다.

Link View List (Explorer 2)

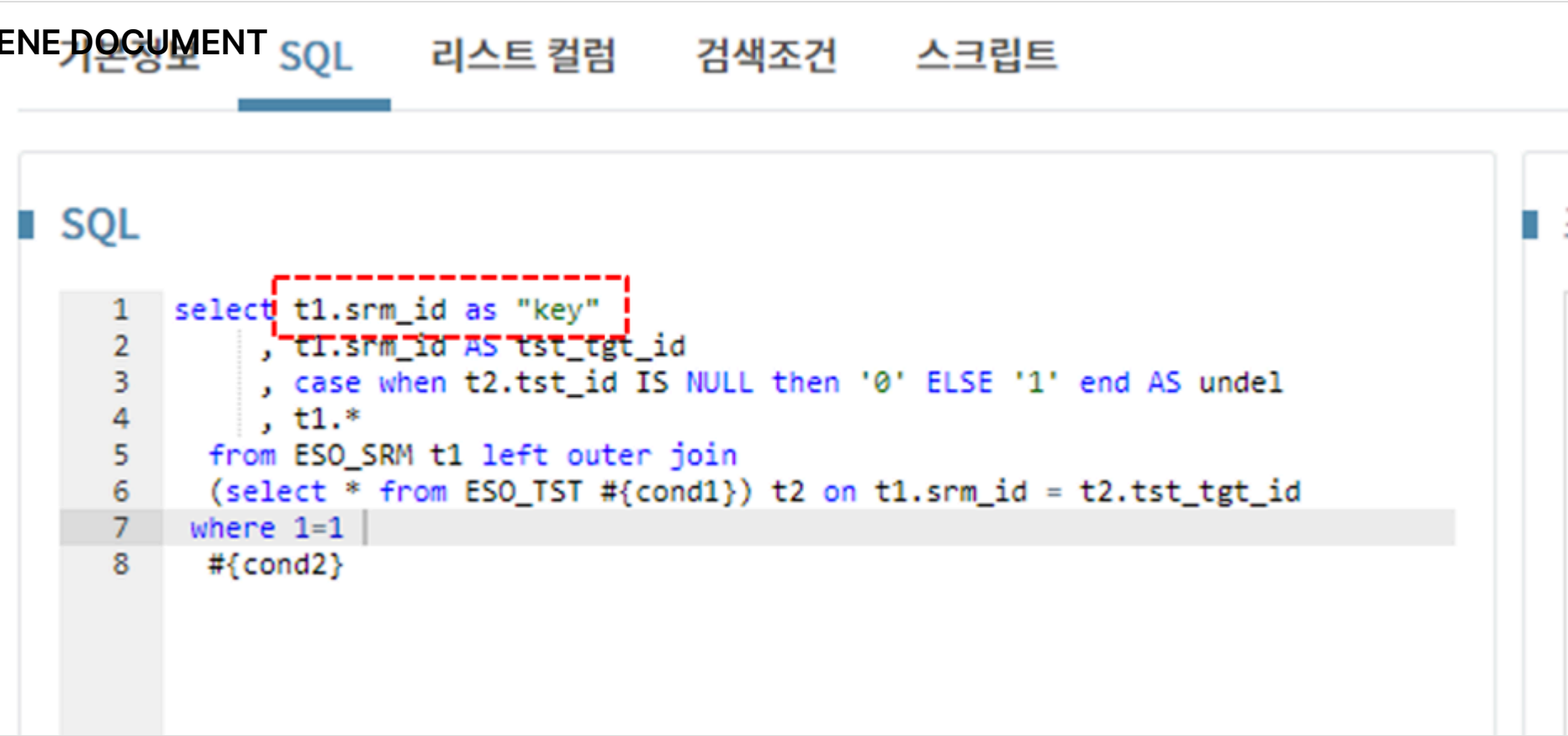
Link View List는 연결 버튼 클릭 시 표시되는 리스트 화면을 구성합니다.

리스트 매니저를 통해 구성하며 이 때 사용하는 유형은 “**연결팝업 유형(Link)**” 입니다.

자세한 리스트 화면 구성은 리스트 매니저 가이드의 리스트 생성하기 목차를 참고바라며, 리스트 유형은 “연결팝업 유형(Link)”입니다.

아래는 리스트를 구성할 때 반드시 설정해야 할 속성에 대한 설명입니다.

1. SQL에 Target Key로 저장할 컬럼을 “key” alias로 설정합니다.



1. 리스트 컬럼에서 선택모드를 checkbox로 설정합니다.



2. 리스트 컬럼 설정

연결팝업 유형의 컬럼 속성은 아래와 같습니다. (* 필수 속성)

속성	설명
* ID	데이터를 보여 줄 컬럼. SQL 쿼리에 작성한 alias와 동일 해야 함.
이름	데이터의 헤더로 표시될 이름.
* 길이	컬럼의 길이.
정렬	정렬 방식(왼쪽, 가운데, 오른쪽)
이벤트 설정	클릭이벤트 설정.
유형	컬럼을 표시할 포맷 유형. 엑셀과 숨김을 지원한다. 엑셀: 화면에서는 해당 컬럼이 숨겨지나 엑셀 다운로드 시 포함된다. 숨김: 화면에서 해당 컬럼이 숨겨지고 데이터만 전달된다.

연결팝업 리스트에서 체크박스로 선택하여 연결할 때(Link 수행) "key" alias로 정의된 값이 함께 전달됩니다.

v6.2 이후 버전에서는 리스트 컬럼에서 컬럼으로 설정한 모든 행 데이터가 ID를 변수명으로 하여 함께 전달됩니다.

API 저장방식에서는 항목설정 따라 변수 값으로 사용할 수 있습니다.

SQL 저장방식에서는 Link 설정에 따라 Arguments 설정의 cols 값으로 사용할 수 있습니다.

Relation 엑셀 업로드 설정 가이드

EGENE DOCUMENT

릴레이션 엑셀 업로드는 일반적인 엑셀 업로드와 다르게 Source ID를 함께 저장해야 해당 릴레이션에 정상적으로 로딩이 됩니다.

로더 저장 방식으로는 ENTITY/SQL 두가지가 있으며, 이 중 ENTITY 저장 방식으로 생성한 로더를 사용할 경우 Source ID(rel_key)가 자동으로 저장됩니다. (단, 해당 폼에 로딩 시 Key 설정이 되어 있어야 합니다.)

SQL 저장 방식의 로더 생성 절차는 복잡하기 때문에 다음의 과정을 따라서 로더를 생성합니다.

로더 생성은 데이터업로드 메뉴에서 할 수 있습니다.

로더 생성 및 업로드 필드 설정

ENTITY 저장 방식

1. 릴레이션 엔터티 선택합니다.

2. [로더 신규 생성] 버튼을 클릭하여 로더를 신규로 생성합니다.

3. 로더 이름 입력합니다.

4. 저장방식을 ENTITY로 설정합니다.

5. 업로드 대상이 되는 필드를 우측으로 이동시켜 로더 필드 설정을 합니다.

6. [로더 필드 속성 변경] 버튼 클릭하여 다음 단계로 이동합니다.

SQL 저장 방식

Home • 시스템관리 • 시스템관리 • 데이터업로드

...

Help

로더 선택

1

sra

SRA[서비스요청관리_릴레이션]

로더 검색

211029_릴레이션테스트[LDR00495:LST03221]

2

로더 신규 생성

로더 정보

로더 아이디

LDR00509

3

로더 이름

릴레이션 로더

4

저장 방식

SQL

엔터티 필드

SRA_ACCOUNT_NAME (계정명)

SRA_ACP_CONTENT (접수내용)

SRA_ACP_DTTM (접수일시)

SRA_ACP_EMP_ID (접수자)

SRA_ACT_CONTENT (처리내용)

SRA_ACT_DTTM (처리일시)

SRA_ACT_EMP_ID (처리자)

SRA_ACTFINISH_DTTM (실제완료일시)

SRA_ACTSTART_DTTM (실제시작일시)

SRA AGREE DTTM (합의일시)

로더필드

5

ID (SRA_ID)

SRA_REQ_EQ_CD (신청 내역)

SRA_USER_ORG_NAME (회사명)

SRA_USER_DPT_NAME (부서명)

SRA_USER_EMP_NAME (실사용자명)

SRA_REQ_REA_CD (요청 사유)

6

로더 필드 속성 변경

1. 릴레이션 엔터티 선택합니다.

2. [로더 신규 생성] 버튼을 클릭하여 로더를 신규로 생성합니다.

3. 로더 이름 입력합니다.

4. 저장방식을 SQL로 설정합니다.

5. 업로드 대상이 되는 필드를 우측으로 이동시켜 로더 필드 설정을 합니다.

6. [로더 필드 속성 변경] 버튼 클릭하여 다음 단계로 이동합니다.

로더 필드 속성 변경

EGENE DOCUMENT

로더 필수 필드 설정 화면에서는 특이사항이 없기 때문에 [NEXT] 버튼을 클릭하여 다음 단계로 이동합니다.

ENTITY/SQL 저장 방식 모두 동일합니다.

Home • 시스템관리 • 시스템관리 • 데이터업로드

BACK

NEXT

column	NAME	필수여부	설명
SRA_ID	ID	<input type="checkbox"/>	
SRA_REQ_EQ_CD	신청 내역	<input type="checkbox"/>	
SRA_USER_ORG...	회사명	<input type="checkbox"/>	
SRA_USER_DPT...	부서명	<input type="checkbox"/>	
SRA_USER_EMP...	실사용자명	<input type="checkbox"/>	
SRA_REQ_REA_CD	요청 사유	<input type="checkbox"/>	

로더 수정 화면에서 SQL 설정

저장방식을 SQL로 설정하면 로더 생성이 완료된 후 로더 수정 화면이 표시되며 이 단계에서 바로 Insert 쿼리를 작성해야 합니다.

SQL 저장 방식일 경우 해당됩니다.

로더 수정

▶ 저장

▶ 삭제

▶ 취소

▶ 닫기

...

기본항목

* 필수입력 항목입니다.

ID

LDR00509

이름

릴레이션 로더

유형

Excel2003~(xlsx)

배치여부

사용여부

1

로더 업로드 타입

개별실행

저장방식

SQL

설명

타겟엔터티ID

[타겟엔터티ID]

연관리스트ID

LST03242

엑셀업로드 샘플파일

+

파일 등록

1. 로더 업로드 타입을 “개별실행”으로 선택합니다. 이에 따라 아래에 INSERT SQL, INSERT SQL ARGS 탭이 표출됩니다.

설정

INSERT_SQL

INSERT_SQL_ARGS

UPDATE_SQL

UPDATE_SQL_ARGS

Post Execute

업로드하기위한 엑셀파일과 매핑을 설정합니다.

2

설정

```
{
  "vars": "",
  "srow": 4,
  "ent": "",
  "row_except_skip": false,
  "delete_first": true,
  "mappings": [
    { "src": "$1", "dest": "s01", "w": 60, "mcol": "sra_id", "alias": "ID ", "require": false },
    { "src": "$2", "dest": "s02", "w": 60, "mcol": "sra_req_eq_cd", "alias": "신청 내역", "require": false },
    { "src": "$3", "dest": "s03", "w": 60, "mcol": "sra_user_org_name", "alias": "회사명", "require": false },
    { "src": "$4", "dest": "s04", "w": 60, "mcol": "sra_user_dpt_name", "alias": "부서명", "require": false },
    { "src": "$5", "dest": "s05", "w": 60, "mcol": "sra_user_emp_name", "alias": "실사용자명", "require": false },
    { "src": "$6", "dest": "s06", "w": 60, "mcol": "sra_req_rea_cd", "alias": "요청 사유", "require": false }
  ]
}
```

2. 설정 값은 로더 필드 설정을 기반으로 자동 구성되어 표시됩니다.

상세항목 ▾

* 필수입력 항목입니다.

설정

INSERT_SQL

INSERT_SQL_ARGS

UPDATE_SQL

UPDATE_SQL_ARGS

Post Execute

데이터 저장 방식이 SQL 이며 , 저장시 동작할 INSERT SQL을 정의하여 사용합니다.

3

INSERT쿼리

```
insert into eso_sra
(sra_id
,sra_src_id
,sra_srm_cat_cd
,sra_req_eq_cd
,sra_user_org_name
,sra_user_dpt_name
,sra_user_emp_name
,sra_req_rea_cd
,sra_reg_dttm
,sra_reg_emp_id

) values (?, ?, 'SRMCAT0501010', ?, ?, ?, ?, ?)
```

3. INSERT SQL 작성 - 아래 "INSERT SQL 및 ARGS 작성 방법"을 참조합니다.

상세항목

설정

INSERT_SQL

INSERT_SQL_ARGS

UPDATE_SQL

UPDATE_SQL_ARGS

Post Execute

데이터 저장 방식이 SQL 이며 , 저장시 동작할 INSERT SQL 의 파라미터를 정의하여 사용합니다.

4

INSERT SQL Args

[
{"cols":"id,rel_key,s02 ,s03 ,s04 ,s05 ,s06 ,cur_dttm , cur_emp"}
]

4. INSERT SQL ARGS 작성 – 아래 “INSERT SQL 및 ARGS 작성 방법”을 참조합니다.

INSERT SQL 및 ARGS 작성 방법

Insert SQL은 엑셀 업로드 데이터를 적용했을 때 수행되는 쿼리이기 때문에 주의해서 입력해야 합니다.

릴레이션 엑셀 업로드를 위해서 반드시 정의해야 하는 필드가 있는데 바로 Source Field입니다. 릴레이션에서는 rel_key라고 불리며 이는 해당 화면 릴레이션을 로딩하기 위한 기본 값이 됩니다.

SQL 저장 방식일 경우 해당됩니다.

아래 예제에서 **sra_src_id**는 rel_key가 저장되는 컬럼이며 rel_key 옵션은 시스템 예약어입니다.

Insert SQL

```
insert into eso_sra (  
    sra_id  
    , sra_src_id  
    , sra_srm_cat_cd  
    , sra_req_eq_cd  
    , sra_user_org_name  
    , sra_user_dpt_name  
    , sra_user_emp_name  
    , sra_req_rea_cd  
    , sra_reg_dttm  
    , sra_reg_emp_id  
) values (?, ?, 'SRMCAT0501010', ?, ?, ?, ?, ?, ?)
```

Insert SQL Args

```
[  
    {"cols":"id,rel_key,s02 ,s03 ,s04 ,s05 ,s06 ,cur_dttm , cur_emp"}  
]
```

INSERT SQL ARS에서 cols에 정의된 예약어중 id는 업로드 데이터의 고유키이며 사용시 업로드에 사용된 유니크키가 저장됩니다.

나머지 컬럼은 설정 탭의 mappings에 정의된 dest, mcol 컬럼을 쌍으로 정의하면 됩니다.

아래는 기타 예약어에 대한 설명입니다.

예약어	설명
id	로더 고유아이디
rel_key	릴레이션 부모 키

EGENE DOCUMENT	현재일시
cur_emp	로그인 사용자 아이디

엑셀 업로드 기능 활성화

ENTITY/SQL 저장 방식 모두 동일합니다.

1. 폼 디자이너에서 Custom.Relation의 엑셀 업로드 체크 활성화

아름

Custom.Relation

값 설정

ID

SRM_PC_PROV_

+

⚙

Edit

☒

Control

삭제

☒

엑셀 다운로드

☐

엑셀 업로드

☒

추가

☒

연결

☐

Args

1. 릴레이션 Config에 엑셀 업로드 설정

설정

엑셀 업로드

☒ 예

☐ 아니오

로더 URI

/xefc/jsp/ui/popup_layout.jsp?uri=/xefc/jsp/loader/loader.jsp&ldr_id=LDR01

```
{
  "layout":{
    "win_w":800,
    "win_h":580
  },
  "upload_uri":"/xefc/jsp/ui/popup_layout.jsp?uri=/xefc/jsp/loader/loader.jsp&ldr_id=LDR00495"
}
```

릴레이션 아톰 리스트

Relation에서 사용할 수 있는 UI 입니다.

Boolean.Checkbox

Boolean.Checkbox는 체크박스 형태의 Atom 입니다.

구분명	Editable
기본	<div><div></div></div>

ui_args 속성

- val0 - value 값이 0일 경우 label 설정. 기본값은 N
- val1 - value 값이 1일 경우 label 설정. 기본값은 Y
- def_val - 기본값 설정. 0 또는 1

예제

```
{
  "id": "text_05",
  "ui_id": "tmp_text_05",
  "alias": "Boolean.Checkbox",
  "w": 100,
  "align": "center",
  "ellipse": true,
  "order": 1,
  "format": null,
  "ui": "Boolean.Checkbox",
  "require": true,
  "readonly": false,
  "changeHandler": "setCheckboxChangeHandlerTest",
  "clickHandler": "setCheckboxClickHandlerTest",
  "clickHandlerArgs": "tmp_text_05",
  "ui_args": [
    "val0":"선택1",
    "val1":"선택2",
    "def_val":"0"
  ]
}
```

Boolean.Radio

Boolean.Radio는 라디오 형태의 Atom 입니다.

구분명	Editable
-----	----------

EGENE DOCUMENT	<div><div><div><div></div><div>N</div></div><div><div>✓</div><div>Y</div></div></div></div>
label 변경	<div><div><div></div><div>선택1</div></div><div><div></div><div>선택2</div></div></div>

ui_args 속성

- val0 - value 값이 0일 경우 label 설정. 기본값은 N
- val1 - value 값이 1일 경우 label 설정. 기본값은 Y
- def_val - 기본값 설정. 0 또는 1

예제

```
{
  "id": "text_07",
  "ui_id": "tmp_text_07",
  "alias": "Boolean.Radio",
  "w": 170,
  "align": "center",
  "ellipse": false,
  "order": 5,
  "format": null,
  "ui": "Boolean.Radio",
  "ui_args": [
    "val0":"실패",
    "val1":"성공",
    "def_val":"1"
  ],
  "readonly": false,
  "require": true,
  "changeHandler": "setRadioHandlerTest"
}
```

String.AutoComplete

String.AutoComplete 는 자동완성 모듈을 사용한 문자열 검색 아톰입니다.

구분명	Editable
기본	<div><div></div><div>✕</div></div>

ui_args 속성

- dic - 사용할 Dictionary 이름
- setvalue - 선택된 값 중 field의value값

value - 선택된 값 중 field의 text값
EGENE DOCUMENT
s_display - 검색 시 호출된 값 목록


- tooltip_set - 검색된 값 Mouse over시 보여줄 내용을 라벨과 같이 지정
- count - 검색 시 최대 호출 개수
- sql_id - 해당 필드의 Readonly시 관련 정보를 조회할 SQL ID
- filters - 검색에 사용할 Filter 값을 | 형태로 구분하여 지정 ex) SVR|HW
- filter_or - Filter에 대한 and/or 사용 여부로 true-and 또는 false-or를 설정. 기본값 false
- wild_card - 텍스트박스 클릭 시 * 와일드카드 동작(전체표출) 사용 여부로 true 또는 false 를 설정. 기본값 false
- mtn_flag - Multitenancy 기능 동작 사용 여부로 true 또는 false를 설정. 기본값 false
- delim - 자동완성 value 구분자
- def_val - 기본값 세팅

예제

```
{
  "id": "s02_nm",
  "alias": "String.AutoComplete",
  "w": 200,
  "align": "center",
  "ellipse": true,
  "order": 3,
  "format": null,
  "ui": "String.AutoComplete",
  "ui_id": "s02",
  "handler": "autoTest",
  "ui_args": [
    {
      "dic": "emp",
      "configParams": "",
      "wild_card": true,
      "filters": "",
      "filter_or": false,
      "def_val": "#{session.user.emp_id}",
      "mtn_flag": true
    }
  ],
  "require": false,
  "readonly": false,
  "changeHandler": "setStringAutoChangeHandlerTest"
}
```

String.Email

String.Email 는 메일 주소 입력 형태의 아톰입니다.

구분명	Editable
기본	

ui_args 속성

없음


EGENE DOCUMENT

예제

```
{
  "id": "text_09",
  "ui_id": "tmp_text_09",
  "alias": "String.Email",
  "w": 300,
  "align": "center",
  "ellipse": true,
  "order": 5,
  "format": null,
  "ui": "String.Email",
  "require": true,
  "changeHandler": "setStringEmailChangeHandlerTest"
}
```

String.Multi.Emp

String.Multi.Emp는 여러 명의 직원을 선택할 수 있는 문자열 형태의 Atom 입니다.

구분명	Editable
기본	

ui_args 속성

없음


예제

```
{
  "id": "text_10",
  "ui_id": "tmp_text_10",
  "alias": "String.Multi.Emp",
  "w": 100,
  "align": "center",
  "ellipse": true,
  "order": 6,
  "format": null,
  "ui": "String.Multi.Emp",
  "require": false,
  "changeHandler": "setStringMultiEmpChangeHandlerTest"
}
```

String.Multi.File

String.Multi.File은 여러 개의 파일을 첨부할 수 있는 첨부파일 형태의 Atom 입니다.

EGENE DOCUMENT

구분명	Editable
기본편집	<div><div></div><div>패치.txt</div><div>작업관리 이력조회</div><div>20210427195940.xlsx</div></div>
읽기	<div><div>패치.txt</div><div>작업관리 이력조회</div><div>20210427195940.xlsx</div></div>

ui_args 속성

ext - 첨부할 파일 확장자 지정, 설정값은 input 태그의 accept 속성값 규칙을 명시한다.

예) 특정파일 확장자 지정 - .docx, .xlsx, .pdf, .txt

오디오, 이미지 비디오 파일의 범위 - audio/* or video/* or image/*

예제

```
{
  "id": "tmp_text_01",
  "alias": "String.Multi.File",
  "w": 100,
  "align": "center",
  "ellipse": false,
  "order": 1,
  "format": null,
  "ui_id": "tmp_text_01",
  "ui": "String.Multi.File",
  "ui_args": {"ext": ".docx,.doc,.xlsx,.xls,.pptx,.ppt,.pdf"},
  "require": false,
  "readonly": false,
  "changeHandler": "setStringMultiFileChangeHandlerTest"
}
```

String.Relation.Key

String.Relation.Key 는 Unique key를 생성해주는 문자열 입력 형태의 Atom 입니다.

구분명	Editable
-----	----------



ui_args 속성

- target_fid - 폼 필드
- params - 전달 파라미터

예제

```
{
  "id": "s03",
  "alias": "String.Relation.Key",
  "w": 100,
  "align": "center",
  "ellipse": false,
  "order": 5,
  "format": null,
  "ui": "String.Relation.Key",
  "ui_args": [
    "F163",
    "rel_ent_id=999&rel_frm_id=FRM000563"
  ],
  "require": false
}
```

String.Single.File

String.Single.File은 하나의 파일을 첨부할 수 있는 첨부파일 형태의 Atom 입니다.

구분명	Editable
기본	
첨부파일 업로드	
읽기상태	

ui_args 속성

- ext - 첨부할 파일 확장자 지정, 설정값은 input 태그의 accept 속성값 규칙을 명시한다.
예) 특정파일 확장자 지정 - .docx, .xlsx, .pdf, .txt
오디오, 이미지 비디오 파일의 범위 - audio/* or video/* or image/*

예제

EGENE DOCUMENT

```
{
  "id": "tmp_text_11",
  "alias": "String.Single.File",
  "w": 100,
  "align": "center",
  "ellipse": false,
  "order": 1,
  "format": null,
  "ui_id": "tmp_text_11",
  "ui": "String.Single.File",
  "ui_args": {
    "ext": ".docx,.doc"
  },
  "handler": "downFile",
  "require": false,
  "readonly": false,
  "changeHandler": "setStringFileChangeHandler"
}
```

String.Text.IP

String.Text.IP는 IP정보를 입력하는 형태의 Atom 입니다.

구분명	Editable
기본	<div><div></div><div></div><div></div><div></div></div>

ui_args 속성

없음


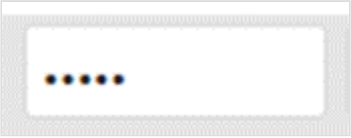
예제

```
{
  "id": "text_06",
  "ui_id": "tmp_text_06",
  "alias": "String.Text.IP",
  "w": 350,
  "align": "center",
  "ellipse": true,
  "order": 2,
  "format": null,
  "ui": "String.Text.IP",
  "require": true,
  "changeHandler": "setIPChangeHandler"
}
```

String.Text

EGENE DOCUMENT

String.Text는 입력하는 형태의 Atom 입니다.

구분명	Editable
기본	
password	

ui_args 속성

- maxleng - 글자 수 입력 제한 길이 값
- type - text input의 타입 설정
- password - 입력값이 표시되지 않고 비밀번호를 입력 받을 수 있다.

예제

```
{
  "id": "text_05",
  "ui_id": "tmp_text_05",
  "alias": "String.Text",
  "w": 250,
  "align": "center",
  "ellipse": true,
  "order": 1,
  "format": null,
  "ui": "String.Text",
  "require": true,
  "changeHandler": "setTextChangeHandler",
  "ui_args": [
    "100"
  ]
}
```

String.TextArea

String.TextArea는 여러 줄의 문자열을 입력하는 형태의 Atom 입니다.

구분명	Editable
기본	

EGENE DOCUMENT

· `__args__` 속성

rows - 텍스트 영역에서 보이는 라인 수


maxleng - 글자 수 입력 제한 길이 값

예제

```
{
  "id": "text_05",
  "ui_id": "tmp_text_05",
  "alias": "String.Text",
  "w": 250,
  "align": "center",
  "ellipse": true,
  "order": 1,
  "format": null,
  "ui": "String.Text",
  "require": true,
  "changeHandler": "setChangeHandlerTest",
  "ui_args": [
    "3",
    "3",
    "200"
  ]
}
```

String.CKEditor

String.CKEditor는 **Custom.MultiSet**에서만 사용 가능한 아톰입니다. Editor를 통해 문자열을 입력하는 형태의 Atom 입니다.

구분명	Editable
기본	

ui_args 속성

rows - 텍스트 영역에서 보이는 라인 수

예제

EGENE DOCUMENT

```
"id": "text_14",
"ui_id": "text_14",
"y": "4",
"x": "1",
"alias": "String.CKEditor",
"w": 300,
"align": "center",
"ellipse": true,
"order": 8,
"format": null,
"ui": "String.CKEditor",
"require": false,
"changeHandler": "setChangeHandlerTest",
"ui_args": [
  "10"
]
}
```

Number.Currency

Number.Currency는 통화 단위로 입력하는 형태의 Atom 입니다.

구분명	Editable
기본	<div></div>
unit 설정	<div>0원</div>

ui_args 속성

unit - 단위 설정

예제

EGENE DOCUMENT

```
"id": "text_06",

"ui_id": "tmp_text_06",

"alias": "Number.Currency",

"w": 250,

"align": "center",

"ellipse": true,

"order": 2,

"format": null,

"ui": "Number.Currency",

"require": true,

"changeHandler": "setChangeHandlerTest",

"clickHandler": "setClickHandlerTest",

"clickHandlerArgs": "tmp_text_06",

"ui_args": [

  "원"

]

}
```

Number

Number는 숫자로 입력하는 형태의 Atom 입니다.

구분명	Editable
기본	<div></div>

ui_args 속성

없음

예제

```
{

  "id": "text_05",

  "ui_id": "tmp_text_05",

  "alias": "Number",

  "w": 250,

  "align": "center",

  "ellipse": true,

  "order": 1,

  "format": null,

  "ui": "Number",

  "require": true,

  "changeHandler": "setNumberChangeHandlerTest",

  "clickHandler": "setNumberClickHandlerTest",

  "clickHandlerArgs": "tmp_text_05"

}
```

EGENE DOCUMENT

Code.Autocomplete

Code.Autocomplete 는 자동완성 모듈을 사용한 문자열 검색 아톰입니다.

구분명	Editable
기본	<div><div></div><div>✕</div></div>

ui_args 속성

- cty_id - 코드유형 값 (릴레이션 유형일때 리스트 fieldType:code 속성이 자동으로 셋팅되며 엑셀다운로드시 코드아이디값을 설정된 코드유형과 치환하여 표기됩니다.)
- dic - 사용할 Dictionary 이름으로 기본값 code
- setvalue - 선택된 값 중 field의 value값
- value - 선택된 값 중 field의 text값
- s_display - 검색 시 표출된 값 목록
- tooltip_set - 검색된 값 Mouse over시 보여줄 내용을 라벨과 같이 지정
- count - 검색 시 최대 표출 개수
- sql_id - 해당 필드의 Readonly시 관련 정보를 조회할 SQL ID
- filters - 검색에 사용할 Filter 값 지정 ex) SVR|HW
- filter_or - Filter에 대한 and/or 사용 여부로 true-and 또는 false-or를 설정. 기본값 false
- wild_card - 텍스트박스 클릭 시 * 와일드카드 동작(전체표출) 사용 여부로 true 또는 false 를 설정. 기본값 false
- mtn_flag - Multitenancy 기능 동작 사용 여부로 true 또는 false를 설정. 기본값 false
- delim - 자동완성 value 구분자
- def_val - 기본값 세팅

예제

EGENE DOCUMENT

```
"id": "s02_nm",

"alias": "Code.Autocomplete",

"w": 200,

"align": "center",

"ellipse": true,

"order": 3,

"format": null,

"ui": "Code.Autocomplete",

"ui_id": "s02",

"ui_args": [

  {

    "dic": "emp",

    "configParams": "",

    "wild_card": true,

    "filters": "",

    "filter_or": false,

    "def_val": "#{session.user.emp_id}",

    "mtn_flag": true

  }

],

"require": false,

"readonly": false,

"changeHandler": "setCodeAutoChangeHandler",

"fieldType" : "code"

}
```

Code.Select

Code.Select 는 Code 정보로 데이터를 구성한 Select box 입력 형태의 Atom 입니다.

구분명	Editable
기본	<div><div>[선택]</div></div>

ui_args 속성

- ctyld - 코드 유형 (릴레이션 유형일때 리스트 fieldType:code 속성이 자동으로 셋팅되며 엑셀다운로드시 코드아이디값을 설정된 코드유형과 치환하여 표기됩니다.)
- restrict - 코드 설정값 설정

예제

EGENE DOCUMENT

```
"id": "txt3",
"alias": "txt3",
"w": 100,
"align": "center",
"ellipse": false,
"order": 10,
"format": null,
"ui": "Code.Select",
"ui_args": [
  "SRMCAT",
  "SRMCAT01010,SRMCAT01020"
],
"require": true,
"changeHandler": "setCodeSelectChangeHandler",
"fieldType" : "code"
}
```

Code.SelectTree

Code.SelectTree 는 Code 정보로 데이터를 구성한 Select box 입력 형태의 Atom 입니다. Select box option은 Tree 형태로 표현된다.

구분명	Editable
기본	<div><div>[선택]</div></div>

ui_args 속성

- subtype - 코드 유형 (릴레이션 유형일때 리스트 fieldType:code 속성이 자동으로 셋팅되며 엑셀다운로드시 코드아이디값을 설정된 코드유형과 치환하여 표기됩니다.)
- restrict - 코드 설정값 설정

예제

```
{
  "id": "tmp_text_04",
  "ui_id": "tmp_text_04",
  "alias": "Code.SelectTree",
  "w": 300,
  "align": "center",
  "ellipse": true,
  "order": 3,
  "ui": "Code.SelectTree",
  "ui_args": [
    "CMCAT"
  ],
  "require": false,
  "changeHandler": "setCodeSelectTreeChangeHandler",
  "fieldType" : "code"
}
```

EGENE DOCUMENT

String.Code.Multi

String.Code.Multi 는 Code 정보로 데이터를 구성한 Select box 의 Multi Check 입력 형태의 Atom이고, Select box option은 체크박스 형태로 표현됩니다. 체크한 데이터는 콤마(,) 구분 형태로 저장됩니다.

구분명	Editable
기본	

ui_args 속성

ctyId - 코드 유형

예제

```
{
  "id": "text_13",
  "alias": "String.Code.Multi view",
  "w": 300,
  "align": "center",
  "ellipse": true,
  "order": 3,
  "ui": "String.Code.Multi",
  "ui_args": {
    "ctyId": "GRADECD"
  },
  "require": false,
  "changeHandler": "setChangeHandler",
  "clickHandler": "setClickHandler",
}
```

Entity

Entity는 기본 Entity 정보로 데이터를 구성한 자동완성 모듈 문자열 검색 아톰입니다.

구분명	Editable
기본	

ui_args 속성

dic : Entity별 사전 정의

EGENE DOCUMENT

setvalue : 선택된 값 중 field의 value값

value : 선택된 값 중 field의 text값

s_display : 검색 시 표출된 값 목록

wild_card : 텍스트박스 클릭 시 * 와일드카드 동작(전체표출) 사용 여부로 true 또는 false 를 설정. 기본값 false

mtn_flag : mtn 기능 사용여부

sql_id:- 해당 필드의 Readonly시 관련 정보를 조회할 SQL ID

tooltip_set : 검색된 값 Mouse over시 보여줄 내용을 라벨과 같이 지정

count : 검색 시 최대 표출 개수

filters : 일반 Text를 필터 사용해야 할 때(and 조건 또는 or조건으로만 필터사용가능, and/or 혼용불가 / #{box.~~} 사용가능)

filter_or : Filter에 대한 and/or 사용 여부로 true-and 또는 false-or를 설정. 기본값 false

delim : 자동완성 value 구분자

hasDelim: true 일때 s_display를 포매팅 형식으로 변경(ex: [V0] V1)

search_target: db , s9s 검색할 대상을 어느 서비스에서 가져올지 선택하기위해 설정 (db: 실시간 검색을 위해 사용합니다)

ent_id - (릴레이션 유형일때 리스트 fieldType:entity 속성이 자동으로 셋팅되며 엑셀다운로드시 아이디값을 설정된 엔터티 마스터테이블 레코드 명과 치환하여 표기됩니다.)

예제

```
{
  "dic": "emp",
  "setvalue": "V0",
  "value": "V1",
  "s_display": "V1,V2",
  "tooltip_set": ["V1|이름", "V0|ID", "V4|회사", "V2|부서"],
  "count": 100,
  "filters": "",
  "filter_or": true,
  "wild_card": true,
  "sql_id": "EntityAuto.Emp.S.01",
  "ui_args": [
    {
      "ent_id": "025"
    }
  ],
  "mtn_flag": false,
  "search_target": "db"
}
```

Ent.Select

Ent.Select 는 Entity 정보로 데이터를 구성한 Select box 입력 형태의 Atom 입니다.

구분명	Editable
기본	<div><div>[Ent.Select]</div><div>▼</div></div>

ui_args 속성

ent_id - 엔터티 ID

예제

EGENE DOCUMENT

```
{
  "id": "tmp_text_14",
  "ui_id": "tmp_text_14",
  "alias": "Ent.Select",
  "w": 300,
  "align": "center",
  "ellipse": true,
  "order": 3,
  "ui": "Ent.Select",
  "ui_args": [
    "044"
  ],
  "require": false,
  "changeHandler": "setEntSelectChangeHandler"
}
```

Entity.Autocomplete

String.Autocomplete 는 자동완성 모듈을 사용한 문자열 검색 아톰입니다.

구분명	Editable
기본	<div><div></div><div>✕</div></div>

ui_args 속성

- dic - 사용할 Dictionary 이름
- setvalue - 선택된 값 중 field의value값
- value - 선택된 값 중 field의 text값
- s_display - 검색 시 표출된 값 목록
- tooltip_set - 검색된 값 Mouse over시 보여줄 내용을 라벨과 같이 지정
- count - 검색 시 최대 표출 개수
- sql_id - 해당 필드의 Readonly시 관련 정보를 조회할 SQL ID
- filters - 검색에 사용할 Filter 값을 | 형태로 구분하여 지정 ex) SVR|HW
- filter_or - Filter에 대한 and/or 사용 여부로 true-and 또는 false-or를 설정. 기본값 false
- wild_card - 텍스트박스 클릭 시 * 와일드카드 동작(전체표출) 사용 여부로 true 또는 false 를 설정. 기본값 false
- mtn_flag - Multitenancy 기능 동작 사용 여부로 true 또는 false를 설정. 기본값 false
- delim - 자동완성 value 구분자
- def_val - 기본값 세팅
- ent_id - (릴레이션 유형일때 리스트 fieldType:entity 속성이 자동으로 셋팅되며 엑셀다운로드시 아이디값을 설정된 엔터티 마스터테이블 레코드 명과 치환하여 표기됩니다.)

예제

EGENE DOCUMENT

```
"id": "s07_nm",
"alias": "Entity.Autocomplete",
"w": 200,
"align": "left",
"ellipse": true,
"order": 2,
"format": null,
"ui": "Entity.Autocomplete",
"ui_id": "s07",
"ui_args": [
  {
    "dic": "emp",
    "wild_card": true,
    "filters": "",
    "filter_or": true,
    "mtn_flag": false,
    "def_val": "#{session.user.emp_id}",
    "ent_id": "025"
  }
],
"require": false,
"readonly": false,
"changeHandler": "setEntAutoChangeHandler",
"fieldType" : "entity"
}
```

Entity.CM.System

Entity.CM.System 은 시스템 정보로 데이터를 구성하고 자동 완성 모듈 및 돋보기 버튼을 클릭하여 데이터를 선택할 수 있는 문자열 입력 형태의 아톰입니다.

구분명	Editable
기본	<div><div></div><div></div></div>

ui_args 속성

ent_id - (릴레이션 유형일때 리스트 fieldType:entity 속성이 자동으로 셋팅되며 엑셀다운로드시 아이디값을 설정된 엔터티 마스터테이블 레코드 명과 치환하여 표기됩니다.)

예제

EGENE DOCUMENT

```
"id": "tmp_text_10",
"ui_id": "tmp_text_10",
"alias": "Entity.CM.System",
"w": 300,
"align": "center",
"ellipse": true,
"order": 3,
"ui": "Entity.CM.System",
"ui_args": [
  {
    "ent_id":"CM"
  }
],
"require": false,
"changeHandler": "setEntCmChangeHandler",
"fieldType" : "entity"
}
```

Entity.Employee

Entity.Employee는 사용자를 선택하는 문자열 입력 형태의 아톰입니다.

구분명	Editable
기본	<div><div></div><div>Q</div></div>

ui_args 속성

ent_id - (릴레이션 유형일때 리스트 fieldType:entity 속성이 자동으로 셋팅되며 엑셀다운로드시 아이디값을 설정된 엔터티 마스터테이블 레코드 명과 치환하여 표기됩니다.)

예제

EGENE DOCUMENT

```
"id": "text_05_1",
"ui_id": "tmp_text_05_1",
"alias": "Entity.Employee",
"w": 400,
"align": "center",
"ellipse": true,
"order": 1,
"format": null,
"ui": "Entity.Employee",
"ui_args": [
  {
    "ent_id": "025"
  }
],
"require": true,
"changeHandler": "setEntEmpChangeHandler"
}
```

ListSearch

ListSearch 는 리스트 데이터를 구성하여 선택하는 문자열 입력 형태의 아톰입니다.

구분명	Editable
기본	<div><div></div><div>Q</div><div>i</div><div>x</div></div>

ui_args 속성

- lst_id - 조회 리스트ID
- info_sql - name 조회 sql
- pop_w - 조회 리스트 Popup 너비. 기본값 800
- pop_h - 조회 리스트 Popup 높이. 기본값 800
- f_params - 전달할 form의 Field ID
- isInfo - 상세 정보 보기 Option
- direct - 직접입력 Option 이다. true 또는 false를 설정. 기본값은 false
- modal_yn - 조회 리스트를 window popup 형태가 아닌 modal popup 형태로 사용할 경우 true로 설정. 기본값은 false(window popup)

예제

리스트 설정

EGENE DOCUMENT

```
"id": "text_08",
"ui_id": "tmp_text_08",
"alias": "Entity.ListSearch",
"w": 300,
"align": "center",
"ellipse": true,
"order": 4,
"ui": "Entity.ListSearch",
"ui_args": {
  "lst_id": "LST_TMPT_002",
  "info_sql": "SQL_TMPT_004",
  "pop_w": "950",
  "pop_h": "600",
  "f_params": "parent.F001,tmp_text_07",
  "isInfo": "true",
  "changeHandler": "setTest"
},
"require": false
}
```

ChangeHandler (릴레이션 초기 스크립트에 함수 정의)

```
$egene.list[listid].setTest = function (row) {
  //row.ui_id : 호출한 ListSearch의 ui_id
  $('<div>relation-atom[uid=' + row.ui_id + ']').parents('<div>.data').find('input[name=tmp_text_34]').val(row.emp_id); //타 필드에 값을 세팅
};
```

조회 리스트 팝업의 스크립트

1) window popup 형태 리스트

```
"script": " var fnCb = opener['fnSetValue_'+ pfid]; var fnCb = opener['fnSetValue_'+ pfid]; fnCb(row.emp_id, row.emp_name, row); self.close(); "
```

2) modal popup 형태 리스트

```
"script": " $egene.dialog.confirm(row.emp_id, row.emp_name, row); "
```

Entity.MPP

Entity.MPP 는 sql로 데이터 구성한 Select box 입력 형태의 아톰입니다. 연결 대상 필드를 지정할 경우, 지정된 필드의 Select box를 재구성할 수 있습니다.

재구성 시, 선택한 필드의 값을 전달하여 sql 데이터를 조회한다.

구분명	Editable
기본	<div><div>[선택]</div></div>

ui_args 속성

sql_id - Select box에 구성할 Sql id

target_col - 연결 필드 ID

primary - 최상위 필드 여부 true 또는 false를 설정. 기본값은 false이다. 최상위 필드일 경우에만 최초 로딩 시 sql 데이터를 조회한다.

EGENE DOCUMENT

예제

```
{
  "id": "text_06",
  "ui_id": "tmp_text_06",
  "alias": "Entity.MPP.01",
  "w": 200,
  "align": "center",
  "ellipse": true,
  "order": 2,
  "ui": "Entity.MPP",
  "ui_args": {
    "sql_id": "MPP.SQL_TMPT_001",
    "target_col": "tmp_text_07",
    "primary": "true"
  },
  "require": true,
  "changeHandler": "setmpp1"
},
{
  "id": "text_07",
  "ui_id": "tmp_text_07",
  "alias": "Entity.MPP.02",
  "w": 200,
  "align": "center",
  "ellipse": true,
  "order": 3,
  "ui": "Entity.MPP",
  "ui_args": {
    "sql_id": "MPP.SQL_TMPT_001"
  },
  "require": true,
  "changeHandler": "setmpp2"
}
```

Sql.Select

Sql.Select 는 sql 정보로 데이터를 구성한 Select box 입력 형태의 아톰입니다. 기존 Entity.Step 기능 중 팝업 설정하는 부분과 파라미터 설정하는 부분이 포함되었습니다.

구분명	Editable
기본	<div><div>[Sql.Select]</div><div>▼</div></div>

ui_args 속성




- sql_id - Select box에 구성할 Sql id
- params - sql args 파라미터 설정이다. {"ID": "값", "ID2": "값2"} 형태로 설정한다.
- popup_url - 팝업으로 보여줄 url 정보 설정

예제

```
{
  "id": "tmp_text_26",
  "ui_id": "tmp_text_26",
  "alias": "Sql.Select",
  "w": 300,
  "align": "center",
  "ellipse": true,
  "order": 3,
  "ui": "Sql.Select",
  "ui_args": {
    "sql_id": "RLT.SQL_TMPT_006",
    "params": {
      "arg1": "test",
      "arg2": "test2"
    }
  },
  "require": false,
  "changeHandler": "setSqlSelectChangeHandler"
}
```

Date.DT

Date.DT는 날짜 선택하는 아톰입니다.

구분명	Editable
기본	<div><div></div><div>2021/05/04</div></div>
현재일 설정버튼 표시	<div><div><div></div><div>2021/05/06</div><div></div></div></div>

ui_args 속성

- currentButton - 현재일 설정 버튼 표시여부로 true 또는 false를 설정. 기본값은 false
- currentInit - 현재일을 기본값으로 설정할지 여부로 true 또는 false를 설정, 기본값은 false
- comp_arr - 설정한 날짜 정합성 체크 기준값을 설정한다.
 - oper - 정합성 체크 비교 연산자 설정. 연산자: <, <=, >, >=, !=, ==
 - target_id - 비교 대상일 정보를 설정. sysdate 또는 대상 컬럼 설정. sysdate는 현재일을 기준으로 비교 수행
 - add_day - 비교 대상일 가감일을 설정. 음수 또는 양수를 설정
 - comp_label - 알림 메시지에 표시할 대상 라벨정보

예제

EGENE DOCUMENT

```
"id": "tmp_text_05",

"ui_id": "tmp_text_05",

"alias": "Date.DT1",

"w": 200,

"align": "center",

"ellipse": true,

"order": 1,

"format": null,

"ui": "Date.DT",

"require": true,

"readonly": false,

"changeHandler": "setDTChangeHandler",

"ui_args": {

  "currentButton": true,

  "comp_arr": [

    {

      "oper": "<",

      "target_id": "tmp_text_13",

      "add_day": "-2",

      "comp_label1": "targetTest"

    },

    {

      "oper": "<=",

      "target_id": "sysdate",

      "add_day": "0"






    }

  ]

}
```

Date.DTTM

Date.DTTM은 일시를 입력하는 아톰입니다. Date.DT 아톰과 구분되는 부분은 시간을 입력할 수 있다는 것입니다.

구분명	Editable
기본	<div><div></div><div></div><div></div><div></div></div>
현재일 설정버튼 표시	<div><div></div><div></div><div></div><div></div><div></div></div>

ui_args 속성

- currentButton - 현재일시 설정 버튼 표시여부로 true 또는 false를 설정. 기본값은 false
- currentInit - 현재일시를 기본값으로 설정할지 여부로 true 또는 false를 설정, 기본값은 false
- comp_arr - 설정한 날짜 정합성 체크 기준값을 설정한다.
- oper - 정합성 체크 비교 연산자 설정. 연산자: <, <=, >, >=, !=, ==

target_id - 비교 대상일 정보를 설정. sysdate 또는 대상 컬럼 설정. sysdate는 현재일을 기준으로 비교 수행

add_day - 비교 대상일 가감일을 설정. 음수 또는 양수를 설정

comp_label - 알림 메시지에 표시할 대상 라벨정보

예제

```
{
  "id": "tmp_text_08",
  "ui_id": "tmp_text_08",
  "alias": "Date.DTTM",
  "w": 300,
  "align": "center",
  "ellipse": true,
  "order": 3,
  "ui": "Date.DTTM",
  "require": false,
  "changeHandler": "setDTTMChangeHandler",
  "ui_args": {
    "currentButton": true
  }
}
```

Date.HH.MM.Select

Date.HH.MM은 시간, 분 값을 입력하는 아톰으로 시는 0~23, 분은 0~~59까지 선택 입력 할 수 있습니다.

구분	화면
기본	<div><div>[시간]▼</div><div>[분]▼</div></div>

ui_args 속성

- hourTerm - 시간 간격 설정. 기본 간격 시간은 1.(예: 2로 설정되면 0, 2, 4, ... 값을 선택)
- minTerm - 분 간격 설정. 기본 간격 분은 1. (예: 2로 설정되면 0, 2, 4 .. 값을 선택)

예제

EGENE DOCUMENT

```
{
  "id": "text_07",
  "ui_id": "text_07",
  "alias": "Date.HH.MM.Select",
  "w": 200,
  "align": "center",
  "ellipse": true,
  "order": 3,
  "ui": "Date.HH.MM.Select",
  "require": true,
  "changeHandler": "setDTTMChangeHandler",
  "ui_args": [
    "1", "1"
  ]
}
```

Date.ST.ET

Date.ST.ET는 기간을 입력하는 아톰입니다.

Editable	Readonly
기본	<div><div><div><div><div></div></div></div><div></div><div>to</div><div></div><div><div><div></div></div></div></div></div>

ui_args 속성

- end_field_id - 종료일 필드 id (sql alias)
- limit - 시작일과 종료일 최대 기간을 숫자로 설정한다.
- comp_arr - 시작일자에 대해 설정한 날짜 정합성 체크 기준값을 설정한다.
 - oper - 정합성 체크 비교 연산자 설정. 연산자: <, <=, >, >=, !=, ==
 - target_id - 비교 대상일 정보를 설정. sysdate 또는 대상 컬럼 설정. sysdate는 현재일을 기준으로 비교 수행
 - comp_label - 알림 메시지에 표시할 비교 대상 라벨정보
 - main_label - 알림 메시지에 표시할 기준 대상 라벨정보
- e_comp_arr - 종료일자에 대해 설정한 날짜 정합성 체크 기준값을 설정한다.
 - oper - 정합성 체크 비교 연산자 설정. 연산자: <, <=, >, >=, !=, ==
 - target_id - 비교 대상일 정보를 설정. sysdate 또는 대상 컬럼 설정. sysdate는 현재일을 기준으로 비교 수행
 - comp_label - 알림 메시지에 표시할 비교 대상 라벨정보
 - main_label - 알림 메시지에 표시할 기준 대상 라벨정보

예제

EGENE DOCUMENT

```
"id": "text_08",
"ui_id": "text_08",
"alias": "Date.ST.ET",
"w": 200,
"align": "center",
"ellipse": true,
"order": 4,
"format": null,
"ui": "Date.ST.ET",
"require": true,
"changeHandler": "setDTTMChangeHandler",
"ui_args": {
  "end_field_id": "tmp_text_09",
  "comp_arr": [
    {
      "oper": ">=",
      "target_id": "sysdate",
      "add_day": "",
      "comp_label": "현재일",
      "main_label": "text_08"
    }
  ]
}
```

Date.YY

날짜 데이터 중 년도를 선택하는 아톰입니다. SelectBox의 UI로 구성됩니다.

구분명	Editable
기본	<div>2021년</div>

ui_args 속성

- minus_y – 현재년도 기준으로 앞으로 보여줄 년도 개수를 설정한다.
- plus_y – 현재년도 기준으로 뒤로 보여줄 년도 개수를 설정한다.

예제

EGENE DOCUMENT

```
{
  "id": "text_10",
  "ui_id": "text_10",
  "alias": "Date.YY",
  "w": 200,
  "align": "center",
  "ellipse": true,
  "order": 5,
  "format": null,
  "ui": "Date.YY",
  "require": true,
  "changeHandler": "setDTTMChangeHandler",
  "ui_args": [
    "3",
    "5"
  ]
}
```

Date.YYMM

날짜 데이터 중 년도와 월을 선택하는 아톰입니다. SelectBox의 UI로 구성됩니다.

구분명	Editable
기본	<div><div>2021년</div><div>▼</div><div>05월</div><div>▼</div></div>

ui_args 속성



tmType - 시분 기본값 설정 구분. 설정값은 0|1|2 한정됨. (0: 000000, 1: 현재 시분초, 2: 235959)

예제

```
{
  "id": "text_11",
  "ui_id": "text_11",
  "alias": "Date.YYMM",
  "w": 200,
  "align": "center",
  "ellipse": true,
  "order": 6,
  "format": null,
  "ui": "Date.YYMM",
  "require": true,
  "changeHandler": "setDTTMChangeHandler",
  "ui_args": [
    "0"
  ]
}
```

Custom.Button
EGENE DOCUMENT

Custom.Button은 이미지 또는 버튼 형태의 아톰입니다. 기존에 사용하던 Custom.Button.Text 와 통합되었습니다.

구분명	Editable
이미지	
텍스트	

ui_args 속성

- type - 버튼 유형 설정. image|text. 기본값 text
- image_src - 버튼 이미지 경로
- label - text 유형의 버튼명. 기본값 필드의 alias
- params - click handler 실행 시 전달할 파라미터이다. {"ID": "값", "ID2": "값2"} 형태로 설정한다. Json handler 함수에서는 row.args1, row.arg2 등으로 호출한다.

예제

```
{
  "id": "tmp_text_11",
  "ui_id": "tmp_text_11",
  "alias": "Custom.Button",
  "w": 100,
  "align": "center",
  "ellipse": true,
  "order": 20,
  "ui": "Custom.Button",
  "clickHandler": "setBtnHandlerTest",
  "clickHandlerArgs": "tmp_text_10",
  "ui_args": {
    "type": "image",
    "image_src": "/images/icon/favorites_on.png",
    "params": {
      "args1": "test",
      "args2": "test2",
      "test3": "test3"
    }
  },
  "require": false
}
```

Custom.File

Custom.File은 파일 첨부하는 아톰입니다. 첨부파일은 1건 업로드가 가능하며 , 재업로드 , 업로드 이력 확인이 가능합니다.

구분명	Editable
-----	----------

ui_args 속성

유의사항

* 물리적인 첨부파일 디렉토리 경로 오류 / 설정된 첨부파일 사이즈 업로드 오류 / 설정된 확장자 외 업로드 오류시에 alert을 통하여 오류 메시지가 나타납니다.

예제

```
{
  "id": "tmp_text_11",
  "alias": "Custom.File",
  "w": 100,
  "align": "center",
  "ellipse": false,
  "order": 1,
  "format": null,
  "ui_id": "tmp_text_11",
  "ui": "Custom.File",
  "require": false,
  "readonly": false
}
```

Relation

Relation 은 릴레이션 유형의 아톰입니다. Custom.MultiSet 리스트에서 사용 가능합니다.

구 분 명	Editable
-------------	----------



판

예제

```
{
  "id": "s99",
  "alias": "Relation",
  "w": 50,
  "align": "center",
  "ellipse": "1",
  "order": 6,
  "ui": "Relation",
  "x": "1",
  "y": "5",
  "cols": "3",
  "ui_args": "{ \"id\": \"SRM_PC_PROV_REL\", \"ctrl_add\": true, \"ctrl_unlink\": true}",
  "require": true,
  "ui_id": "s99",
  "format": "hide"
}
```

릴레이션 활용하기-리스트 UI

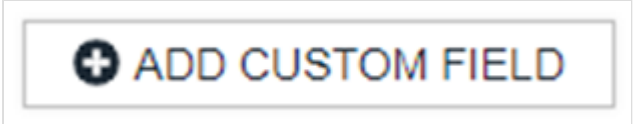
릴레이션 정보는 관리자메뉴 관계관리 화면에서 관리하며 이렇게 정의된 릴레이션은 폼 디자이너에서 1:N 또는 N:N의 데이터를 구성하기 위한 아톰으로 사용할 수 있습니다.
여기에서 리스트 UI 형태로 화면을 구성하기 위한 아톰, Custom.Relation을 설명합니다.



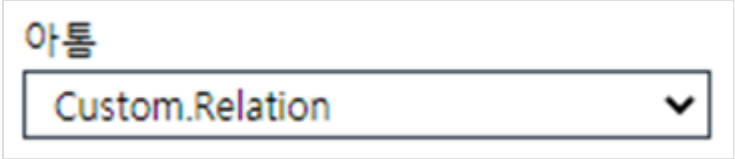
Custom.Relation 설정하기

릴레이션은 관계정보에 의해 렌더링 되는 아톰이기 때문에 Custom 필드로 추가해야 합니다.
다음은 폼 디자이너에서 릴레이션 아톰을 추가하고 정의하는 순서입니다.

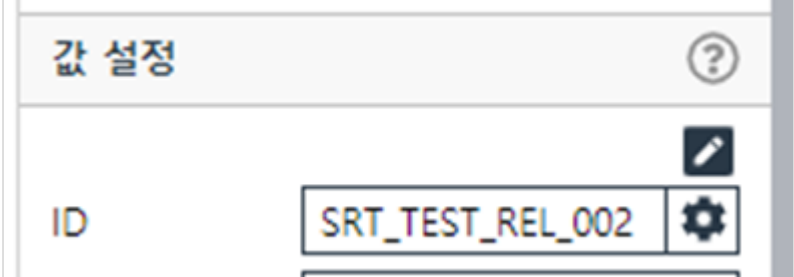
1. 폼 디자이너에서 릴레이션이 위치할 Set을 선택하고 상단 [ADD CUSTOM FIELD] 버튼을 이용하여 Custom Field 추가



2. Field의 속성창에서 아톰 항목을 Custom.Relation로 선택



3. 값 설정 패널의 ID 항목에 릴레이션 아이디 등록



4. 릴레이션 아이디 설정 후 미리보기 항목 확인

EGENE DOCUMENT

txt1	txt2	txt3	txt4	txt5	txt6	txt7	txt8	txt9

릴레이션 아이디 설정 후에 미리보기에 설정한 내용이 표시되지 않으면 릴레이션 정보에 문제가 있기 때문에 설정 내용을 다시 확인해야합니다.

5. 릴레이션 아토를 선택했다면 값 설정 패널에서 릴레이션 아토 속성을 설정한다.

아토 속성

아토의 전체 속성을 설명하고 각 속성의 자세한 내용은 다음 이어지는 내용에서 알아보겠습니다.

아토

Custom.Relation

값 설정

?

1 ID

SRM_SE_DOMA

+

⚙

2 Edit

☒

Control

3 삭제

☒

4 엑셀 다운로드

☐

5 엑셀 업로드

☐

6 추가

☒

7 연결

☐

8 Args

1. ID: 릴레이션 ID 설정
2. Edit: 릴레이션의 editable 편집상태를 설정하며 설정 값은 (true/false)로 설정합니다.

true: 설정된 아토를 화면에 표시합니다.
false: 데이터만 표시되며 편집은 불가한 상태입니다.
3. 삭제: 릴레이션 항목 삭제 기능 사용 여부 선택

삭제 선택 시 체크박스 선택된 항목을 삭제 처리한다.
4. 엑셀 다운로드: 릴레이션 엑셀 다운로드 버튼 사용 여부 선택
5. 엑셀 업로드: 릴레이션 엑셀 업로드 기능 사용 여부 선택
6. 추가: 릴레이션 항목 추가 기능 사용 여부 선택

추가 선택 시 설정된 내용에 따라 리스트 행추가 또는 입력폼 팝업을 이용하여 추가한다.
7. 연결: 릴레이션 연결 기능 사용 여부 선택

연결은 연결 대상이 되는 데이터 목록을 리스트로 표시하여 선택적으로 매핑하게 된다.
8. Args: Link(추가) 쿼리, Apply(수정) 쿼리, 릴레이션 추가 화면에 전달할 값을 명시합니다.

ex) arg1=value&arg3=#{box.[parameterName]}&arg4=#{row.[columnName]}
9. create_label (JSON 설정): 추가 버튼 라벨을 설정
10. excel_create_label (JSON 설정): 엑셀 업로드 버튼 라벨 설정
11. link_label (JSON 설정): 연결 버튼 라벨 설정

추가

폼 디자이너에서 추가를 체크하면 릴레이션 항목을 추가하는 작업을 할 수 있습니다. 기본 기능은 릴레이션 리스트에 새로운 행(row)을 추가하여 데이터를 편집할 수 있도록 합니다.

릴레이션 config 설정에 따라 데이터가 없을 때 기본 행을 보여주거나 폼화면을 이용한 추가, 다른 입력 화면을 이용한 추가를 할 수 있습니다.

저장방식을 API로 설정 시 릴레이션 마법사 또는 매니저 화면을 이용하게 되며 해당 설정은 설정 탭에서 수정할 수 있습니다.

EGENE DOCUMENT

1. 기본 행 사용 기능 설정

추가 기본 기능인 행추가 기능일 때 설정할 수 있으며 데이터가 없을 때 기본으로 하나의 빈 행을 보여주는 옵션입니다.
릴레이션 config 설정에서 autoAddRow 속성으로 설정합니다. (true/false)

■ 설정

추가 유형

행

기본 행 사용

☒ 예

☐ 아니오

```
{
  "autoAddRow": false
}
```

2. 폼화면을 이용하여 릴레이션 항목 추가

폼화면 팝업을 이용한 추가기능을 이용하고자 할 때 설정합니다.
릴레이션 config 설정에서 addType을 form으로 설정하고 form 속성에 폼 아이디 설정합니다.
두 속성 모두 존재해야 폼 팝업 추가 유형을 사용할 수 있습니다.

■ 설정

추가 유형

폼 팝업

폼

```
{
  "addType": "form",
  "form": "FRM004690"
}
```

폼화면에서 추가 할 경우 저장후 릴레이션 리스트를 조회하게 됩니다. 이때 삭제나, 수정중인 데이터가 초기화 되기 때문에 폼 화면 추가 전에 저장해야 합니다.

3. 특정화면을 이용하여 추가

특정 URI를 팝업 하여 추가기능을 이용하고자 할 때 설정합니다.
릴레이션 config 설정에서 addType을 url로 설정하고 create_uri 속성에 URI 설정합니다.
두 속성 모두 존재해야 URI 팝업 추가 유형을 사용할 수 있습니다.



```
{
  "addType": "url",
  "create_uri": "/xefc/jsp/ui/form/form.jsp?ent_id=CM&frm_id=FRM003497&act=1"
}
```

폼 릴레이션 항목 추가화면 개발 시 릴레이션 갱신을 고려해야 합니다. 팝업 화면 오픈 시 relation uid를 전달하며 uid를 이용하여 relation 갱신을 수행해야 합니다.
샘플) opener.updatebounddataList.call(null, undefined, relUid);

삭제 EGENE DOCUMENT

릴레이션 항목 삭제 기능으로, 삭제 항목들은 폼 저장 시 일괄 삭제됩니다.

연결속성을 사용하거나 폼 추가 유형이 행추가가 아닐 경우 추가 행위가 실시간 저장이기 때문에 삭제 또한 실시간 삭제됩니다.

연결

연결은 기존에 존재하는 정보를 릴레이션을 통해서 연결하는 작업입니다. 연결은 리스트 화면에서 연결 대상을 선택하여 수행하게 되며, 연결 리스트는 Link View List에 설정된 리스트가 표시됩니다.

엑셀 다운로드

릴레이션 데이터를 엑셀 다운로드하는 작업입니다. 폼디자이너에서 엑셀 다운로드 속성을 체크하여 설정합니다.

확장자 : .xlsx

엑셀 업로드

엑셀 업로드 모듈을 이용하여 릴레이션 데이터를 추가하는 작업입니다. 폼디자이너에서 엑셀 업로드 속성을 체크 후 릴레이션 config를 설정합니다.

저장방식을 API로 설정 시 릴레이션 마법사 또는 매니저 화면을 이용하게 되며 해당 설정은 설정 탭에서 수정할 수 있습니다.

1. 로더 설정

릴레이션 config 설정에서 upload_uri 속성에 로더 ID를 포함한 URI를 설정합니다.

설정

엑셀 업로드

☒ 예 ☐ 아니오

로더 URI

/xefc/jsp/ui/popup_layout.jsp?uri=/xefc/jsp/loader/loader.jsp&ldr_id=LDR01

```
{
  "layout":{
    "win_w":800,
    "win_h":580
  },
  "upload_uri": "/xefc/jsp/ui/popup_layout.jsp?uri=/xefc/jsp/loader/loader.jsp&ldr_id=LDR01"
}
```

릴레이션에서 사용하는 로더 설정은 일반적인 로더 설정과 다르므로 Relation 엑셀 업로드 설정 가이드를 참조하여 생성해야 합니다.

Args (파라미터)

폼디자이너에서 args로 설정한 파라미터를 실제 DML로직이나 화면으로 전달하기 위해서는 릴레이션 config에서 파라미터를 추가해줘야 한다.

저장방식을 API로 설정 시 릴레이션 마법사 또는 매니저 화면을 이용하게 되며 해당 설정은 설정 탭에서 수정할 수 있습니다.

1. 전달 파라미터 설정

릴레이션 config 설정에서 args속성에 파라미터 변수명을 , 구분자로 입력합니다.

EGENE DOCUMENT

설

파라미터

arg1,arg2,arg3

```
{
  "args": "arg1,arg2,arg3"
}
```

Custom Button 추가하기

릴레이션 기본 처리기능외에 추가 버튼 기능이 필요한 경우 처리 버튼을 추가할 수 있습니다.
Custom Button은 Relation List에서 설정하며, Relation List script에 정의된 함수를 호출하게 됩니다.

1. List 설정 중 controls 속성으로 버튼 정의

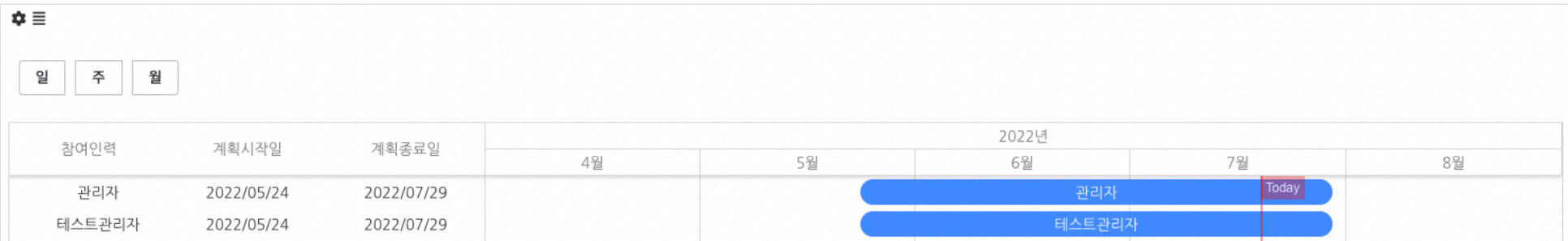
```
{
  ...
  "controls": [
    {
      "alias": "버튼이름",
      "handler": "handlerFunction"
    }
  ]
}
```

2. Script 설정에 hander function 정의

```
{
  "scripts": [
    {
      "name": "handlerFunction",
      "script": "alert('custom');"
    }
  ]
}
```

Custom.Gantt 설정하기

Custom.Gantt은 간트 차트 아톰입니다.



EGENE DOCUMENT

폼 설정

- id: 릴레이션 아이디
- frm_id: row 클릭하여 수정할 때 사용하는 폼 및 새로운 row를 추가할 때 사용할 폼 (필수 x)
- ent_id: 엔터티 아이디
- ctl_id: 저장 후 수행 될 제어 버튼의 아이디
- args: 넘길 동적 값 (ex. pms_id=#{row.pms_id})
- edit: 수정 여부
- startDateField: 시작일
- endDateField: 종료일
- textField: 제목
- empField: 사용자(직원) 자동완성 필드
- wogField: 작업그룹 자동완성 필드
- zoomType: 기본 타임라인 날짜 단위 (day/week/month/year)

예제

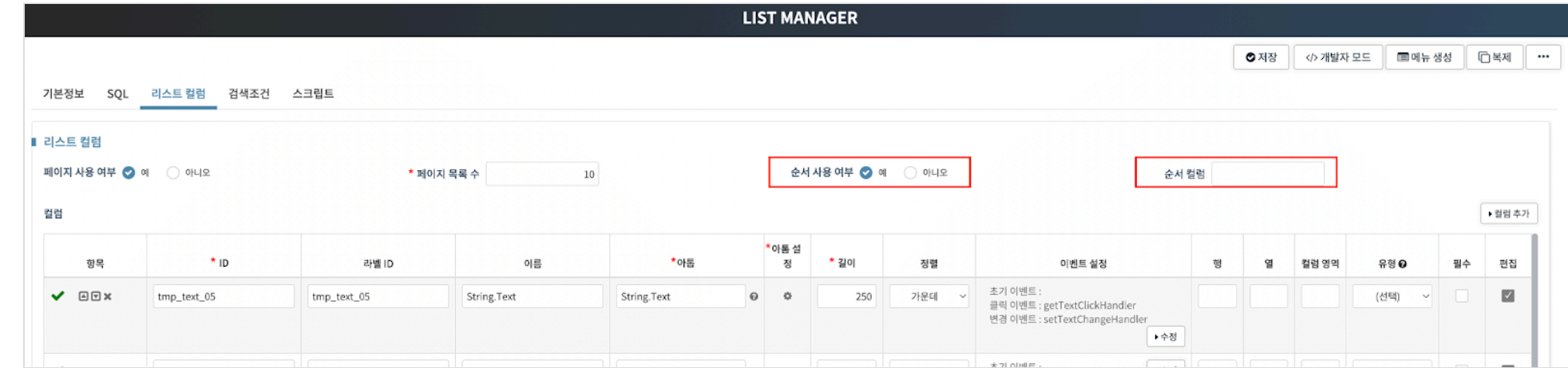
```
{
  "id": "릴레이션 ID",
  "frm_id": "",
  "ent_id": "",
  "ctl_id": "",
  "args": "",
  "edit": false,
  "startDateField": "planstart_dttm",
  "endDateField": "planfinish_dttm",
  "textField": "emp_name",
  "zoomType": "month"
}
```

Relation 순서 기능 사용하기

Relation row들을 원하는 순서로 정렬하고 싶을 때 유용한 기능입니다.

버튼으로 손쉽게 순서를 조정할 수 있어 사용하기 쉽습니다.

사용방법



릴레이션 리스트에 순서 사용 여부를 예로 변경 후 해당 순서를 저장할 컬럼 이름을 적어주세요.

이때 컬럼 이름은 릴레이션 Link 설정의 cols에 넣은 이름과 동일해야하고

상세정보

EGENE DOCUMENT

DML

insert into eso_template(tmp_id, tmp_text_05,tmp_order,tmp_src_id, tmp_text_12)
values(7,7,7,RTMPCATV_ORDER)

Argument

{
 "cols":"#key, tmp_text_05,tmp_order,#rel_key",
 "require":"#key,#rel_key"
}

SQL에 orderby에 순서가 저장된 컬럼 (ex. tmp_order)을 설정해주어야합니다.

SQL

```
1 select tmp_id "key",  
2     tmp_text_05,  
3     tmp_text_05 as text_05  
4 from eso_template  
5 where tmp_text_12 = 'RTMPCATV_ORDER' -- Boolean  
6 #ifcond)  
7 order by tmp_order
```

릴레이션 순서 기능이 활성화 되면 다음과 같이 표시됩니다.

☰

▶ 추가

▶ 삭제

제목

릴레이션 활용하기-폼 필드 UI

릴레이션 정보는 관리자메뉴 관계관리 화면에서 관리하며 이렇게 정의된 릴레이션은 폼 디자이너에서 1:N 또는 N:N의 데이터를 구성하기 위한 아톰으로 사용할 수 있습니다.

여기에서 폼 필드 UI 형태로 화면을 구성하기 위한 아톰, Custom.MultiSet을 설명합니다.

MultiSet은 추가 버튼을 이용하여 설정된 필드를 폼 화면 형태로 반복적으로 추가하여 1:N 형식의 데이터를 등록 할 수 있게 합니다.

▶ Add

*일자

✕ Delete

B I U S

☰

문단

글꼴

크기

A A

Performance

*일자

✕ Delete

B I U S

☰

문단

글꼴

크기

A A

Performance

Custom.MultiSet 설정하기

릴레이션은 관계정보에 의해 렌더링 되는 아톰이기 때문에 Custom 필드로 추가해야 합니다.

다음은 폼 디자이너에서 Custom.MultiSet 아톰을 추가하고 정의하는 순서입니다.

1. 폼 디자이너에서 릴레이션이 위치할 Set을 선택하고 상단 [ADD CUSTOM FIELD] 버튼을 이용하여 Custom Field 추가

EGENE DOCUMENT

ADD CUSTOM FIELD

2. Field의 속성창에서 아톰 항목을 **Custom.MultiSet**으로 선택

아톰

Custom.MultiSet

3. 값 설정 패널의 config 항목에 아톰 고유 속성을 JSON Object 형태로 입력

config

```
{
  "rlt_id": "TMPRLT021_02",
  "rel_key": "#
  {box.key}",
  "ctrl_add": true,
  "ctrl_unlink":
```

아톰속성

속성명	설명
rlt_id	릴레이션 아이디를 설정합니다.
rel_key	릴레이션 구조 아톰은 참조키를 기본으로 설정해야 합니다. 기본으로 “#{box.key}”로 설정하며 현재 화면의 키 값을 전달하도록 합니다.
edit	데이터 등록을 사용할지 여부 설정이며 기본값은 false입니다. true 설정 시 입력 컴포넌트로 구성됩니다.
ctrl_add	추가 버튼 사용여부이며 true 설정 시 추가 버튼이 표시됩니다.
create_label	추가 버튼의 라벨을 설정합니다. 선택사항으로 설정하지 않으면 “추가”를 기본 값으로 표시됩니다.
args	MultiSet 아톰에 전달할 값을 명시합니다. ex) arg1=value&&arg3=#{box.[parameterName]}&arg4=#{row.[columnName]}

샘플

```
{
  "rlt_id": "TMPRLT021_02",
  "rel_key": "#{box.key}",
  "edit": true,
  "ctrl_add": true,
  "args": "",
  "create_label": ""
}
```

MultiSet에 특화된 List 설정하기

MultiSet은 릴레이션 구성 설정을 사용하지만 List 설정의 속성이 차이가 있습니다.

MultiSet의 List 구성은 개발자 모드에서만 가능하며 List설정 값을 제외하고 릴레이션(explorer) 유형의 List설정 방법과 동일합니다.

MultiSet의 Relation List (Explorer 1)는 List설정에 layout과 grid를 정의해야 합니다.

```
{
  "layout": {
    "field_cols": 2,
    "field_lw": 120
  },
  "grid": {
    "fields":[
      {
        "id": "s01",
        "alias": "Text 1",
        "ui": "String.Text",
        "require": true,
        "x": "1",
        "y": "1",
        "cols": "1"
      },
      ...
    ]
  }
}
```

Layout 설정

MultiSet 폼 화면의 공통 레이아웃 설정입니다.

속성명	설명
field_cols	폼 필드 칼럼 개수를 설정합니다. 정의한 수만큼 폼 칼럼(열이) 생성됩니다. ex) 1열: 1, 2열: 2
field_lw	폼 라벨 길이를 설정하며 단위는 px이고 정수 값만 설정합니다. 기본값은 120입니다.

grid의 fields 설정

grid는 폼 화면을 구성하는 필드 정보를 설정합니다. fields는 필드 목록을 정의하며 배열구조로 설정합니다.

속성명	설명
id	필드 아이디로 sql alias, 등록 수정 칼럼 아이디와 동일해야 합니다.
alias	폼 필드에 표시되는 이름입니다.
ui	폼 화면을 구성하는 아토름을 설정합니다. 아토름은 릴레이션 아토름과 동일합니다.
require	필수 여부 설정으로 true 또는 false를 설정한다. 기본값은 false입니다.
x	폼 위치 정보이며 행 위치 값이 됩니다. 왼쪽부터 1로 시작하도록 설정합니다.
y	폼 위치 정보이며 열 위치 값이 됩니다. 위에서부터 1로 시작하도록 설정합니다.

릴레이션 활용하기 - 기타

메일 콘텐츠에 릴레이션 정보도 포함하기 위한 설정 방법입니다.

메일 콘텐츠 설정하기

Content 설정

시스템관리 > Content 설정 메뉴에서 Content 메일 본문에 릴레이션 ID를 설정합니다.

설정 방법 : #{relation : relation ID}

The screenshot shows a web browser window displaying an email body template. The top part has a blue header bar with the placeholder text '#{system.title}'. Below this is a white section titled '요청정보' (Request Information). Inside this section is a table with two columns: field names and their corresponding placeholders. The fields are ID, 단계 (Stage), 요청일시 (Request Date), 요청분류 (Request Category), 제목 (Title), 내용 (Content), and 릴레이션 (Relation). The '릴레이션' field's value, '#{relation:SRM_PC_RENT_REL}', is enclosed in a red rectangular box. At the bottom left of the page, there is a blue hyperlink labeled '링크 바로가기' (Link Directly).

ID	#{req_id}
단계	#{tas_name}
요청일시	#{req_dttm}
요청분류	#{cat_name}
제목	#{req_title}
내용	#{req_content}
릴레이션	#{relation:SRM_PC_RENT_REL}

[링크 바로가기](#)

관계 관리

시스템관리 > 관계 관리 메뉴에서 릴레이션 엔터티 ID를 설정합니다.

기본정보

* ID

SRM_PC_PROV_REL

이름

서비스요청_지급/교체

엔터티 ID

SRA

Explorer 1

서비스요청_지급/교체 요청

🔍 ... ✕

❗ 릴레이션 리스트에서 보여주는 Explorer 유형의 리스트를 지정한다.

E-GENE Lighting (Dev)				
요청정보				
ID	SR2208-00745			
단계	접수			
요청일시	2022-08-29 16:18			
요청분류	대여 요청			
제목	대여 요청 테스트			
내용	대여 요청 테스트			
릴레이션	신청내역	실 사용자	사용기간	요청사유
	기타	관리자	20220829~20220830	출장

릴레이션 연결 - 리스트 선택 유형

릴레이션 연결은 연결대상이 되는 항목을 조회하여 선택하는 방식으로 작동합니다.

연결대상 모수가 많을 경우 검색이 필수이기 때문에 검색 선택하는 방식이 최선일 수 있습니다.

모수가 적을경우 예를 들면 권한처럼 모수가 한정된 데이터를 연결할 경우 전체 목록을 표시하여 연결 상태만 체크하면 사용자 편의성이 증대됩니다.

rel_jqxtreegrid.jsp 연결 대상을 표시하여 선택하는 방식으로 작동하는 릴레이션 연결 컴포넌트입니다.

저장

조회

ID	name	pid
▼ <input checked="" type="checkbox"/> doc	웹문서 관리	ROOT
<input checked="" type="checkbox"/> DOCMEN00769	공유 문서함	doc
<input type="checkbox"/> DOCMEN00770	나의 문서함	doc

특징

- jqxtreegrid를 활용하여 계층형 데이터 표시 가능
- 연결대상만 선택하여 저장
- 저장 버튼을 클릭시 DB 저장
- 조회 버튼을 클릭 시 데이터 갱신

컴포넌트

URL: /lit/rel_jqxtreegrid.jsp

parameter

- sync: 연결정보 동기화 여부
- rlt_id: 릴레이션 아이디
- auth: 수정 권한
- rel_key: 부모 키

릴레이션 개발 가이드

아톰 개발 가이드

릴레이션 각 셀은 아톰으로 구성되며 특정 아톰 설정이 없다면 `String.Text`로 구성됩니다.

`String.Text`는 편집상태에서 텍스트를 입력받을 수 있는 기본 텍스트 박스가 표시되는데 특별한 기능이 필요한 편집 아톰이 필요할 경우 아톰 개발이 필요합니다.

아래 코드는 릴레이션 아톰의 기본 템플릿으로 개발이 필요한 경우 기본 템플릿을 구성하고 아래 가이드에 따라서 개발하면 됩니다.

아톰 템플릿

```
/xefc/jsp/ui/relation/uiitem/Basic.jsp
```

EGENE DOCUMENT

```
<?<pageContent type="text/html; charset=utf-8" %>

<%@ include file="Field.Common.jsp" %>

<%

    // JSP 표현식

%>

<%

    if (editable) {

        // 편집모드 수행

    %>

<%

    // 아톰 로딩시 실행되는 즉시실행함수

    (function () {

        // Relation Atom

        var $atom = $('<relation-atom[uid=<uid%>]>'),

            uiid = $atom.attr('uiid');

        <!-- change Handler 설정 -->

        var $inputVal = $('<input[name='+uiid+']>', $atom),

            $inputTxt = $('<input[name='+uiid+'_txt']>', $atom);

        $inputTxt.on('change', function(event){

            // 저장할 값 찾아오기

            var val = $(this).val();

            // hidden에 값 세팅

            $inputVal.val(val);

            // 변경된 값 row에 반영

            var relUid = $('<#<= fid %>>').attr('uid');

            var key = $atom.parents('<div.data>').attr("key");

            var row = $egene.list[relUid].rows[key];

            row.atom = $atom;

            row.readonly = false;

            row[uiid] = $inputVal.val();

            // change handler 수행

            var func = $egene.list[relUid]['<= f.changeHandler %>'];

            if (typeof func == 'function') {

                func.call(null, row);

            }

            // 상태 업데이트

            setChecked($(this));

        });

    })();

    /**
```

EGENE DOCUMENT

```
* 데이터 적합성 체크
* @returns {boolean}

*/

function relation_validate_<%= uid %>() {

    return true;

}


/**
* 데이터 필수 체크
* @returns {boolean}
* @private
*/

function relation_check_require_<%= uid %>() {

    return true;

}


<%
} else {

    out.println("");

    if (f.hasClickHandler()) {

        out.println("" + prefix + val + suffix + "");

    } else {

        out.println(prefix + val + suffix);

    }

    out.println("");

%>


(function () {

    // Relation Atom

    var $atom = $('<div>.relation-atom[uid=<%=uid%>]</div>');

    <!-- click Handler 설정 -->

    var $linkObj = $('<div>.link', $atom);

    $linkObj.on('click', function () {

        // row 데이터 조회

        var relUid = $('<div>#<%= fid %></div>').attr('uid');

        var key = $atom.parents('<div>.data').attr("key");

        var row = $egene.list[relUid].rows[key];

        row.atom = $atom;

        row.readonly = <%= readonly %>;

        // click handler 수행

        var func = $egene.list[relUid]['<%= f.clickHandler %>'];

        if (typeof func == 'function') {

            func.call(null, row);

        }

    });

})();

<%
}

%>
```



```
<%@ include file="Field.Common.jsp" %>
```

Field.Common.jsp는 아톰 필드의 공통 영역으로 데이터 처리나 아톰 상태 처리를 기본으로 정의하고 있습니다.

아톰 root 태그

```
" class="relation-atom atom-group <%= cls %>" uuid="<%=f.ui_id%>">
```

아톰의 시작을 알리는 시작 root 태는 uid, uuid 속성을 가지고 있으며 relation-atom 클래스명을 가지고 있습니다.

uid 아톰이 생성될때마다 발급되는 유니크한 아이디가 설정되고, uuid는 해당 컬럼의 필드 아이디가 설정됩니다.

JQuery Selector를 이용하여 릴레이션 아톰을 찾는 경우 relation-atom 클래스명을 이용해야 합니다.

편집모드 제어

```
<%  
    if (editable) {  
        // 편집모드 수행  
    }  
>
```

editable는 릴레이션 아톰이 생성되는 시점에 편집상태 여부 값을 가지고 있으며 true 값이면 편집 아톰을 화면에 표시합니다. 릴레이션 아톰이 편집상태이고, 컬럼이 편집상태라면 true, 릴레이션 아톰이나 컬럼중 하나가 편집상태가 아니면 false 값을 가집니다. readonly 상태는 데이터 수정 여부이며 readonly 상태가 아니라고 해서 편집할 수 있는 것은 아닙니다.

```
// 아톰 로딩시 실행되는 즉시실행함수

(function () {

    // Relation Atom

    var $atom = $('<div>relation-atom[uid=<%=uid%>]</div>', $atom),

        uiid = $atom.attr('uiid');

    //

    var $inputVal = $('<input type="text" name="+uiid+">', $atom),

        $inputTxt = $('<input type="text" name="+uiid+"_txt">', $atom);

    $inputTxt.on('change', function(event){

        // 저장할 값 찾아오기

        var val = $(this).val();

        // hidden에 값 세팅

        $inputVal.val(val);

        // 변경된 값 row에 반영

        var relUid = $('<div>#<%= fid %></div>').attr('uid');

        var key = $atom.parents('<div>div.data</div>').attr("key");

        var row = $egene.list[relUid].rows[key];

        row.atom = $atom;

        row.readonly = false;

        row[uiid] = $inputVal.val();

        // change handler 수행

        var func = $egene.list[relUid]['<%= f.changeHandler %>'];

        if (typeof func == 'function') {

            func.call(null, row);

        }

        // 상태 업데이트

        setChecked($(this));

    });

})();
```

릴레이션 아톰별로 즉시 실행 함수가 수행되며 uid를 이용하여 아톰 제어를 수행합니다. 소스 가독성을 위해서 JSP 표현식의 사용을 권장하지 않으나 사용에 제한은 없습니다. 기본 샘플은 uid를 이용하여 릴레이션 컬럼 \$atom을 가져오고 있으며, \$atom을 scope로 활용하여 아톰 구성요소를 제어합니다. change handler 설정도 해당 영역에서 제어합니다.

```
/**
 * 데이터 정합성 체크
 * @returns {boolean}
 */
function relation_validate_<%= uid %>() {
    return true;
}

/**
 * 데이터 필수 체크
 * @returns {boolean}
 * @private
 */
function relation_check_require_<%= uid %>() {
    return true;
}
```

validation 함수는 데이터 정합성을 체크하는 함수이며 저장 시 기본 호출됩니다. require 함수는 데이터 등록 여부를 체크하는 함수로 require 체크되어 있는 컬럼은 저장 시 호출됩니다.

일반모드

```
out.println("");
if (f.hasClickHandler()) {
    out.println("" + prefix + val + subfix + "");
} else {
    out.println(prefix + val + subfix);
}
out.println("");
```

일반모드는 편집상태가 false 상태를 말하며, 데이터만 표시합니다.

기본 구성은 clickHandler 설정 여부에 따라서 class에 link를 설정하며 표시는 데이터만 합니다. 실제 Handler는 아톰 실행 함수(일반모드)에서 정의합니다. 해당 부분은 데이터 표시 방식에 따라서 추가 확장하면 됩니다.

```
// 아톰 로딩시 실행되는 즉시실행함수

(function () {

    // Relation Atom

    var $atom = $('<div>relation-atom[uid=<%=uid%>]</div>');

    //

    var $linkObj = $('<div>.link', $atom);
    $linkObj.on('click', function () {

        // row 데이터 조회

        var relUid = $('<div>#<%= fid %>', $atom).attr('uid');
        var key = $atom.parents('div.data').attr("key");
        var row = $egene.list[relUid].rows[key];

        row.atom = $atom;

        row.readonly = <%= readonly %>;

        // click handler 수행

        var func = $egene.list[relUid][<%= f.clickHandler %>];

        if (typeof func == 'function') {

            func.call(null, row);

        }

    });

})();
```

일반모드 화면에 clickHandler 설정 여부에 따라서 핸들러를 호출합니다.

아톰 기본 속성

속성명	설명
box	릴레이션 Request Parameter Map
uid	릴레이션 아톰 고유 아이디
_user	로그인 사용자 정보
texts	다국어 라이브러리
row	릴레이션 현재 행 데이터 오브젝트
idx	릴레이션 컬럼 순서
f	ListField 오브젝트 컬럼 정보로 상세 내용은 다음 챕터에서 설명
val	저장 데이터
style	css 스타일 정의
etc	리스트 기타 설정 정보
fid	릴레이션 아톰 필드 아이디
readonly	읽기 여부 상태
editable	편집 여부 상태

EGENE DOCUMENT	아톰 클래스 정의
strRO	readonly 상태 문자 값, readonly일때 "readonly"
strDS	disabled 상태 문자 값, disabled일때 "disabled"
prefix	읽기모드 출력 문자열 접두사
subfix	읽기모드 출력 문자열 접미사

ListField

ListField는 리스트 필드 객체이며 기본적으로 리스트 필드 구성정보로 구성됩니다. 리스트 필드 설정 정보를 처리하고자 한다면 ListField 값을 참고해야 합니다.

속성

속성명	설명
id	설정된 아이디는 화면에 표시될 데이터 아이디를 지정 ui_id 생략시 id값을 ui_id로 자동 설정됨.
ui_id	편집시 저장될 아이디로, 정의되지 않았다면 id와 동일하다.
alias	컬럼이름으로 헤더에 표시되는 이름
w	필드 가로길이
cls	스타일 클래스 이름
etc	root 태그에 저장할 속성
align	문자열 정렬 [left center right]
ellipse	긴 문자열 자동 개행 여부
changeHandler	변경 핸들러 이름

메소드

속성명	설명
getArgument(String name)	리스트 설정에서 ui_args가 오브젝트인 경우 이름으로 값을 가져온다.
getArgument(int index)	리스트에 설정된 ui_args가 배열인 경우 배열 인덱스로 값을 가져온다.
getDefVal	필드 기본값

스크립트 API

릴레이션에서 사용 가능한 스크립트 사용법입니다.

EGENE DOCUMENT API	설명	버전
fnGetCurrentRow	선택된 Row 정보 조회	6.1
fnGetCurrentRowValue	선택된 Row에서 필드 값을 조회	6.1
fnSetCurrentRowValue	전달된 Row에 Input 속성의 아톰 값을 설정	6.1
fnSetSelectCurrentRowValue	전달된 Row의 SelectBox 유형의 아톰 값을 설정	6.1
doRefresh	릴레이션 새로고침	
checkRelationAll	릴레이션 전체 체크	
getRelationInfo	릴레이션 설정정보 조회	
dmlRelation	릴레이션 DML 수행	
addRelationItem	빈 row 추가	
addRelationItemX	데이터 기준 row 추가	
showRelButton	릴레이션 제어버튼 표시	
hideRelButton	릴레이션 제어버튼 숨김	
showRelAllButton	릴레이션 제어버튼 전체 표시	
hideRelAllButton	릴레이션 제어버튼 전체 숨김	

fnGetCurrentRow (v.6.1)

전달된 Row값을 통하여 정보를 돌려줍니다.

@param row 선택된 릴레이션 row의 data 객체

```
var $rData = fnGetCurrentRow(row);
```

fnGetCurrentRowValue (v6.1)

전달된 Row에서 필드 값을 조회합니다.

@param \$rData 선택된 릴레이션 row의 html 객체

@param col_name data 조회할 컬럼 명

```
var val = fnGetCurrentRowValue($rData, 'emp_id');
```

fnSetCurrentRowValue (v6.1)

전달된 Row에 Input 속성의 아톰 값을 설정합니다.

@param \$rData 선택된 릴레이션 row의 html 객체

@param col_name data 설정할 컬럼 명

@param val 설정 값

EGENE DOCUMENT

```
// row.info_text 값을 Info Text Input 아톰에 설정
fnSetCurrentRowValue($rData, 'info_text', row.info_text);
```

fnSetSelectCurrentRowValue (v.6.1)

전달된 Row의 SelectBox 유형의 아톰 값을 설정합니다.

@param \$rData 선택된 릴레이션 row의 html 객체

@param col_name data 설정할 컬럼 명

@param val 설정 값

```
// row.emp_id 값을 emp_id 아톰에 설정
fnSetSelectCurrentRowValue ($rData, 'emp_id', row.emp_id);
```

doRefresh

선택한 릴레이션을 새로고침 합니다. 이 때 조회에 필요한 파라미터를 전달할 수 있습니다.

@param fid 릴레이션 필드 ID (ex. Field_F001)

@param params 릴레이션을 재 조회 할 때 전달할 파라미터 {k:v}로 설정

```
doRefresh('Field_F003',{test: 'aaa'});
```

checkRelationAll

릴레이션의 데이터 행을 전체 체크하거나 체크 해제합니다.

@param fid 릴레이션 필드 ID (ex. Field_F001)

@param b 체크 상태, true: 체크 | false: 체크 해제

```
checkRelationAll('Field_F003', true);
```

getRelationInfo

릴레이션의 설정정보를 조회하여 돌려줍니다.

@param rlt_id 릴레이션 ID

```
getRelationInfo('RLT_TMPT_001');
```

dmlRelation

릴레이션의 설정 정보와 데이터에 따라 등록/수정/삭제를 수행합니다.

@param rlt_config 릴레이션 config 정보

@param gubun dml구분(i/u/d)

@param items dml처리 할 데이터 목록

@param ent_id Entity ID

@param reL_key 릴레이션 연결key

EGENE DOCUMENT

```
//delete 실행
var rlt_config = getRelationInfo('RLT_TMPT_101');

var result = dmlRelation(rlt_config,

    'd',

    [{key:'165465426200034'}],

    '999',

    'TMP2206-00195'

);
```

addRelationItem

릴레이션에 데이터가 없는 빈 행을 추가합니다.

@param *fid* 릴레이션 필드 ID (ex. Field_F001)

```
addRelationItem('Field_F003');
```

addRelationItemX

릴레이션에 전달받은 데이터로 setting한 행을 추가합니다.

@param *fid* 릴레이션 필드 ID (ex. Field_F001)

@param *item* 추가한 행에 보여 질 데이터

```
addRelationItemX('Field_F003', {

    tmp_text_05:'VSVR030',

    tmp_text_06:'SVR010',

    tmp_text_07:'SVR010'

});
```

showRelButton

릴레이션에 특정 제어버튼을 동적으로 표시합니다.

@param *fid* 릴레이션 필드 ID (ex. Field_F001)

@param *id* 버튼 ID

```
showRelButton('Field_F003','add');
```

버튼 ID는 고정되어 있으며 다음과 같습니다.

- 추가: add
- 삭제: unlink
- 엑셀 업로드: all_upload
- 엑셀 다운로드: excel
- 리스트 custom 버튼: custom_[index] (index: custom 버튼의 index로 0부터 시작한다.)

hideRelButton

릴레이션에 특정 제어버튼을 동적으로 숨깁니다.

@param *fid* 릴레이션 필드 ID (ex. Field_F001)

@param *id* 버튼 ID

EGENE DOCUMENT

```
hideRelButton('Field_F003','add');
```

버튼 ID는 고정되어 있으며 다음과 같습니다.

추가: add

삭제: unlink

엑셀 업로드: all_upload

엑셀 다운로드: excel

리스트 custom 버튼: custom_[index] (index: custom 버튼의 index로 0부터 시작한다.)

showRelAllButton

릴레이션에 전체 제어버튼영역을 동적으로 표시합니다.

@param *fid* 릴레이션 필드 ID (ex. Field_F001)

```
showRelAllButton('Field_F003');
```

hideRelAllButton

릴레이션에 전체 제어버튼영역을 동적으로 숨깁니다.

@param *fid* 릴레이션 필드 ID (ex. Field_F001)

```
hideRelAllButton('Field_F003');
```