inswave systems

WRM(WebSquare5 Reference Model) for WebSquare5 SP5

UI 개발 표준 정의서

2022.06.16

WebSquare5

재개정 이력

문서 버전	작성일자	작성자	설명
0.1	2018-03-09	박상규	문서 작성 시작
1.0	2021-03-19	박상규	W-Docs 기능을 활용한 소스 설명 주석 작성 방법 업데이트 및 컴포넌트 식별자 규칙 추가
1.1	2023-06-16	박상규	cm 디렉터리 하위에 추가된 디렉터리(template, udc)에 대한 설명 추가
1.2	2024-06-17	박채원	디렉토리 변경 (WEBSQUARE_AI_DEV_PACK)

목 차

1. 디렉터리 구조 및 파일 생성 규칙

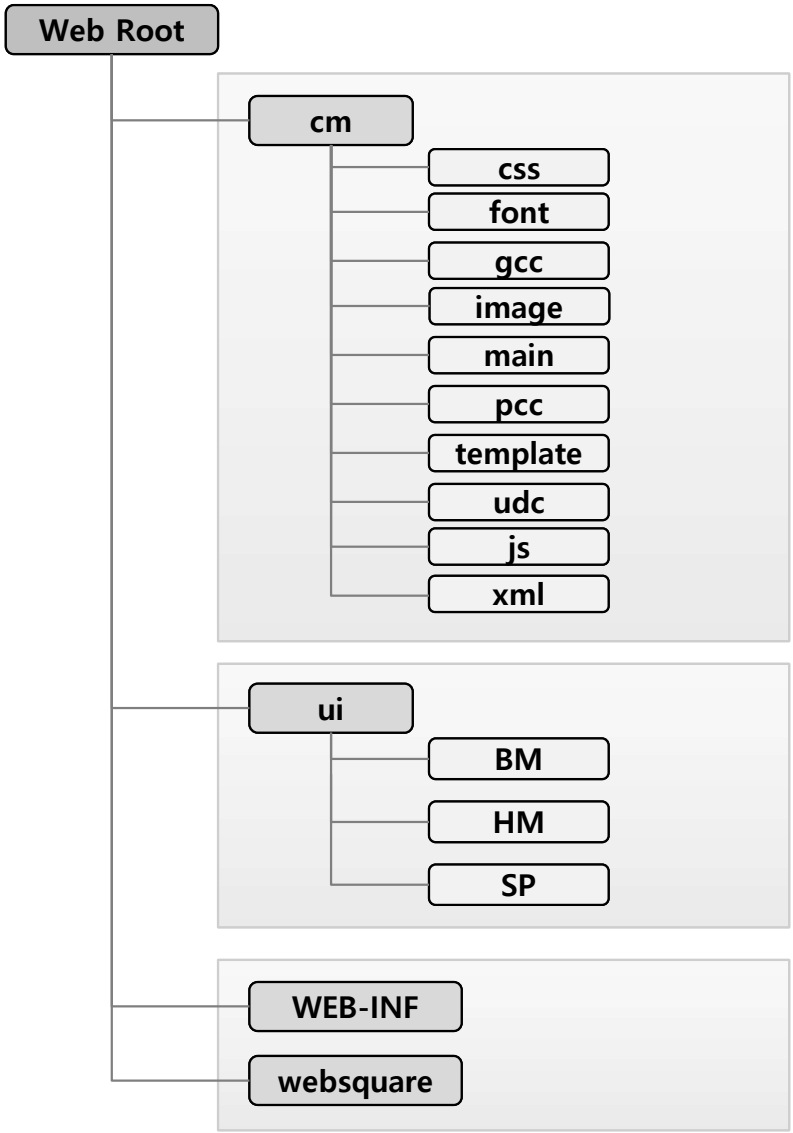
1. 1. 디렉터리 구조
1. 2. 공통 리소스 명명 규칙
1. 3. 업무 UI 파일 명명 규칙
1. 4. 샘플 UI 파일 명명 규칙

2. UI 코딩 표준

2.1. 컴포넌트 명명 규칙
2.2. 변수 명명 규칙
2.3. 함수 명명 규칙
2.4. 자바 스크립트 코딩 표준

1. 1. 디렉터리 구조

UI 개발을 생성되는 리소스 파일들을 저장하기 위한 디렉터리 구조 정의



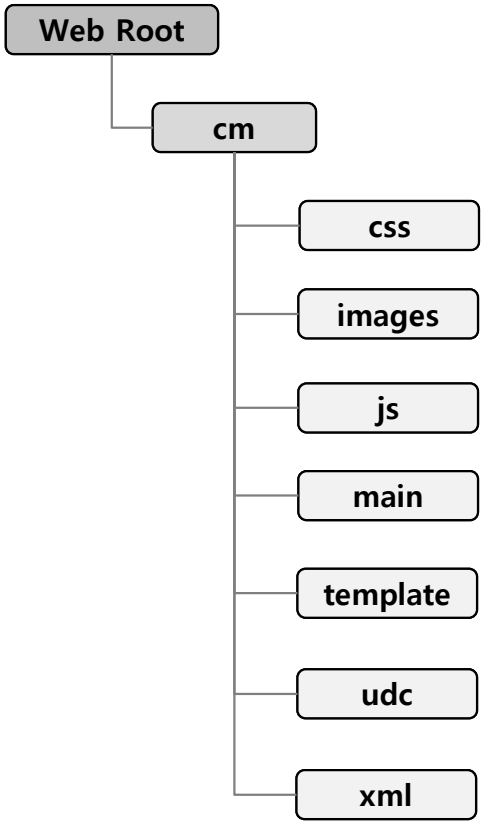
cm	• 공통 css, font, javascript 파일을 저장하는 디렉터리의 상위 디렉터리
+ css	• 공통 css 파일
+ font	• 공통 font 파일
+ gcc	• 공통 함수가 작성된 Common Component 파일
+ images	• 화면 내에서 사용하는 이미지 파일들을 저장. 이미지의 사용 용도에 따라서 하위 디렉터리를 생성해서 저장할 수 있음
+ js	• External JS 등이 필요한 경우 사용하는 파일
+ main	• 공통 함수에 대한 Javascript Document 파일 (화면 로딩 시 불필요)
+ pcc	• Page Common Component 파일
+ template	• 화면 퍼블리싱을 위한 Template 파일 (화면 로딩 시 불필요)
+ udc	• 사용자 정의 컴포넌트(UDC) 파일
+ js	• 공통 함수 및 업무 공통 함수 파일
+ xml	• 메인 화면에서 Include하는 Side 메뉴, 메인 화면 레이아웃 설정 팝업 • 업무 화면 공통 Header, Footer 소스 파일과 공통 메지시 창, 팝업 창 공통 소스 파일

ui	• 웹스퀘어로 개발된 모든 UI 소스 파일(XML)을 저장 • ui 디렉터리 안에 들어가는 업무별 디렉터리(BM, HM)의 숫자가 많거나 UI 소스 파일의 숫자가 많다면, 업무 별 디렉터리 아래에 하위 디렉터리를 만들어서 저장할 수도 있음
+ BM	• 공통 코드, 권한, 프로그램, 메뉴, 릴리즈, 파일그룹, 단축키 관리 화면
+ HM	• 조직 정보 관리 및 인사 기본 정보 관리 화면
+ SP	• UI 공통 함수 및 웹스퀘어 컴포넌트의 기능을 사용 방법 설명을 위한 샘플 화면

WEB-INF	• 웹스퀘어 엔진 및 관련 JAVA Library, Web & Spring Framework 설정 파일
websquare	• 웹스퀘어 엔진에서 사용하는 리소스 및 설정 파일

1. 2. 공통 리소스 파일 명명 규칙

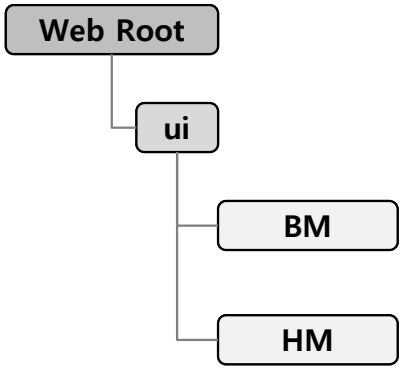
공통 리소스(css, font, js, image), UI 공통 파일을 생성하기 위한 파일 명명 규칙 정의



css	**명명규칙**	• 파일 확장자 : css • 파일명 허용 문자 : 영문 소문자, '_' • 각 시스템 별 필요에 의해 별도의 스타일이 필요한 경우에는 css 파일내의 클래스명으로 구분하거나, 업무별로 스타일이 완전히 달라지는 경우에는 별도의 css 파일을 분리할 수 있음
	적용사례	• all.css, main.css, responsible.css
images	**명명규칙**	• 파일 확장자 : gif, jpg, png • 파일명 허용 문자 : 영문 소문자, 숫자, '_' • 전 업무에서 공통으로 사용하는 image는 image의 모양을 알 수 있는 명칭을 사용하도록 해야 함 • 각 업무별 이미지는 <어플리케이션 코드 소문자>_image명으로 명명함
	적용사례	• 공통 이미지 : insert.gif, delete.gif, sub_title.gif • 업무별 이미지 : kq_sub_title.gif, ks_insert.gif
js	**명명규칙**	• 파일 확장자 : js • 파일명 허용 문자 : 영문 소문자, '_' • 공통적으로 사용하는 스트립트나 업무별 공통 스크립트는 별도의 js 파일로 분리해서 관리함. 업무별 개별 화면의 스크립트는 웹스퀘어 소스(XML) 파일에 inline으로 작성
	적용사례	• 공통 스크립트 : commonGlobal.js, commonScope.js • 메인 화면 스크립트 : main.js • 다국어 메시지 스크립트 : en.js, ko.js
udc **main** **xml**	**명명규칙**	• 파일 확장자 : xml • 파일명 허용 문자 : 영문 대소문자, 숫자, '_'
	참고사항	• 메인 화면에서 Include하는 Side 메뉴, 메인 화면 레이아웃 설정 팝업 소스 • 사용자 정의 컴포넌트(UDC) 파일 소스 • 업무 화면 공통 Header, Footer 소스 파일과 공통 메지시 창, 팝업 창 공통 소스 파일
	적용사례	• header.xml, footer.xml, side.xml, main.xml, login.xml, messageBox.xml

1. 3. 업무 UI 파일 명명 규칙

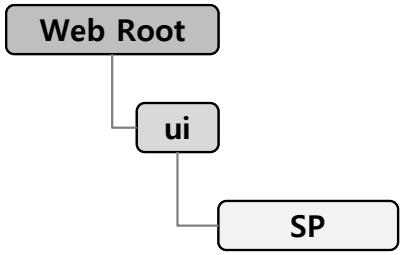
개별 업무 UI 파일(XML)을 생성하기 위한 파일 명명 규칙 정의



<table>
<tr><td rowspan="3">업무 UI</td><td>명명규칙</td><td>• 파일 확장자 : xml
• 파일명 허용 문자 : 영문 대문자, 숫자, '_'
• 파일명 생성 규칙
•
<table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td></tr></table>• [1~2] : 어플리케이션 디렉터리명(어플리케이션 코드(2자리) + 알파벳(1자리))
• [3~5] : 단위업무 메인 화면 일련번호(숫자 3자리)
• [6] : 화면 구분 (M : 메인화면, P : 팝업화면, S : 서브(Wframe) 화면)
• [7~8] : [1~6]를 기준으로 한 일련번호(숫자 2자리)</td></tr>
<tr><td>참고사항</td><td>• 개별 업무 화면 UI 소스 파일 (웹스퀘어 XML 파일)</td></tr>
<tr><td>적용사례</td><td>• BM001M01.xml : 공통업무 첫번째 단위 업무 메인 화면
• BM002M01.xml : 공통업무 두번째 단위 업무 메인 화면
• BM002P01.xml, BM002P02.xml : 공통업무 두번째 단위 업무 메인 화면에서 사용되는 팝업 화면
• BM002F01.xml, BM002F02.xml : 공통업무 두번째 단위 업무 메인 화면에서 사용되는 서브(Wframe) 화면</td></tr>
</table>

1. 4. 샘플 UI 파일 명명 규칙

개발 샘플 및 기능 샘플에 대한 폴더 및 파일 명명 규칙



샘플 UI	명명규칙	• 폴더/파일명 허용 문자 : 영문자, 숫자, '_' • 폴더명 생성 규칙 : - 컴포넌트의 기능인 경우 컴포넌트명 - 컴포넌트가 없는 경우 주요 기능 단어 • 파일명 생성 규칙 : - 컴포넌트명 + 기능명 - 기능명
	참고사항	• 업무 화면이 아닌 기능에 관한 샘플 화면들에 대한 명명 규칙.
	적용사례	• GridView/GridView_Excel_LargeData.xml : 대량 데이터를 그리드뷰에 표현 예시 화면 • InputBox/InputBox_DisplayFormat.xml : InputBox 컴포넌트의 데이터 포맷 적용 예시 화면

2. 1. 컴포넌트 아이디 명명 규칙

UI 소스(웹스퀘어 XML) 파일 안에서 사용되는 컴포넌트의 아이디 명명 규칙 정의

명명규칙	• 컴포넌트 아이디 : <컴포넌트 식별자>_<컴포넌트명> • 컴포넌트 식별자 : 컴포넌트의 종류 식별을 위해서 컴포넌트 아이디 맨 처음에 작성함. 영문 소문자 3자리 • 컴포넌트명 허용 문자 : 영문 대소문자, 숫자 • 컴포넌트명는 소문자를 기본으로 사용함 • 컴포넌트명의 첫번째 문자는 소문자로 시작함 • 2개 이상의 단어가 연결되는 경우에는 단어의 첫 글자를 대문자로 작성함 • 컴포넌트 아이디에 사용되는 단어는 의미 있는 이름을 사용해야 함 • 컴포넌트 아이디를 통해서 저장될 컴포넌트의 사용 용도를 추측할 수 있어야 함
참고사항	• UI 개발 툴에서 컴포넌트 생성 시 자동으로 아이디(컴포넌트 식별자 + "_" + 컴포넌트 기본 아이디 + 일련번호, ex. btn_input1)를 생성함 • 자바스크립트 코드에서 참조하는 컴포넌트들은 반드시 컴포넌트 아이디 명명 규칙에 맞도록 컴포넌트 아이디를 재정의해야 함
적용사례	• acb_searchKeyword • ica_startDate1 • ibx_custNo

> 컴포넌트 식별자

컴포넌트명	식별자	설명	컴포넌트 아이디 명명 예제
AutoComplete	acb	자동완성 셀렉트박스	acb_searchKeyword
AliasDataList	adl	부모 WFrame의 DataList의 Alias 생성	adl_billSection
AliasDataMap	adm	부모 WFrame의 DataMap의 Alias 생성	adm_billMst
Trigger, Anchor	btn	버튼	btn_save, btn_insert
InputCalendar	ica	날짜 입력용 inputBox	ica_startDate
Calendar	cal	달력	cal_applcDate
CheckBox	cbx	체크박스	cbx_abolYn

컴포넌트명	식별자	설명	컴포넌트 아이디 명명 예제
CheckCombobox	ccb	체크박스콤보	ccb_acptTpCd
Chart, FusionChart	cht	Chart, FusionChart	cht_aplyStat1
DataList	dlt	리스트 형태의 Multi Row 데이터 저장	dlt_billSection
DataMap	dma	맵 형태의 단건 Row 데이터 저장	dma_billMst
DatePicker	dpk	DatePicker	dpk_billYmd
Editor	edt	에디터	edt_reqDetail
FloatingLayer	flt	FloatingLayer	flt_custMst
Fliptoggle	ftg	Fliptoggle	ftg_selBillYn
Generator	gen	Generator (특정 UI 요소를 반복 생성)	gen_menuType
Grid, GridView	grd	Grid, GridView	grd_billSection
Group	grp	그룹	grp_custMst
Iframe	ifm	Iframe	ifm_billDetail1
Image	img	이미지	img_subTitle
InputBox	ibx	입력박스	ibx_custNo
LinkedDataList	ldt	링크드데이터리스트	ldt_acptTypeCd
Menu	mnu	메뉴	mnu_dlMenu
Multiupload	mpd	파일 멀티 업로드(플래쉬 사용)	mpd_attachFile
MultiSelect	msb	멀티셀렉트박스	msb_selBillCd
Output	opt	아웃풋박스	opt_aplySeq

컴포넌트명	식별자	설명	컴포넌트 아이디 명명 예제
PageControl	pgc	페이지 컨트롤	pgc_billMst
PageList	pgl	페이지 리스트	pgl_custMst
Radio	rad	라디오박스	rad_selBillCd
Submission	sbm	서버와 통신 기능 제공	sbm_custMstInsert
SelectBox	sbx	셀렉트박스	sbx_atmp
Secret	sct	입력한 문자가 보이지 않는입력박스	sct_password
SlideHide	shd	SlideHide	shd_menu
Spinner	spi	Spinner	spi_aplySeq
span	spn	Span	spn_authorityCnt
Switch	swh	스위치(Switch)	swh_billSection
Tab	tab	탭컨트롤 안에 각 탭	tab_person
TabControl	tac	탭컨트롤	tac_rearch
TableLayout	tbl	테이블레이아웃	tbl_billDetail
Tag	tag	태그	tag_needle
TreeView	trv	트리뷰	trv_orgInfo
Textarea	txa	Textarea	txa_billDetail
Textbox	tbx	텍스트박스	tbx_title
Upload	upd	1개의 파일 업로드	upd_attach1
WindowContainer	wdc	MDI	mdi_main
WFrame	wfm	WFrame (JSP Include와 같이 동작)	wfm_custDetail1

※ 컴포넌트 명명 규칙 적용 예제 코드 (DataList, DataMap, Submission)

```
<xf:model>
    <xf:instance>
        <data xmlns="" />
    </xf:instance>
    <w2:dataCollection baseNode="map">
        <w2:dataList id="dlt_commonSearchItem" baseNode="list" saveRemovedData="true" repeatNode="map">
            <w2:columnInfo>
                <w2:column id="GRP_NM" name="그룹명" dataType="text"></w2:column>
                <w2:column id="GRP_CD" name="그룹코드" dataType="text"></w2:column>
                <w2:column id="NOTE" name="비고" dataType="text"></w2:column>
            </w2:columnInfo>
        </w2:dataList>
        <w2:dataMap baseNode="map" id="dma_resultAll">
            <w2:keyInfo>
                <w2:key id="STATUS" name="STATUS" dataType="text"></w2:key>
                <w2:key id="ICNT_GRP" name="코드그룹_입력건수" dataType="text"></w2:key>
                <w2:key id="ICNT_CODE" name="코드_입력건수" dataType="text"></w2:key>
                <w2:key id="UCNT_GRP" name="코드그룹_수정건수" dataType="text"></w2:key>
                <w2:key id="UCNT_CODE" name="코드_수정건수" dataType="text"></w2:key>
                <w2:key id="DCNT_GRP" name="코드그룹_삭제건수" dataType="text"></w2:key>
                <w2:key id="DCNT_CODE" name="코드_삭제건수" dataType="text"></w2:key>
                <w2:key id="TYPE" name="TYPE" dataType="text"></w2:key>
            </w2:keyInfo>
        </w2:dataMap>
    </w2:dataCollection>
    <w2:workflowCollection></w2:workflowCollection>
    <xf:submission id="sbm_commonCode" ref='data:json,dma_CommonGrp' target='data:json,dlt_commonCode' action="/common/selectCommonCodeList"
        method="post" ediatype="application/json" encoding="UTF-8" errorHandler="" customHandler="" mode="asynchronous" processMsg=""
        ev:submitdone="scwin.sbm_commonCode_submitdone" ev:submiterror="" abortTrigger="">
    </xf:submission>
    <xf:submission id="sbm_commonGroup" ref='data:json,dma_Search' target='data:json,dlt_commonGrp' action="/common/selectCommonGroup"
        method="post" mediatype="application/json" encoding="UTF-8" errorHandler="" customHandler="" mode="asynchronous" processMsg=""
        ev:submitdone="scwin.sbm_commonGroup_submitdone" ev:submiterror="" abortTrigger="">
    </xf:submission>
</xf:model>
```

※ 컴포넌트 명명 규칙 적용 예제 코드 (Group, inputBox, gridView, Anchor)

```
<body ev:onpageload="scwin.onpageload" ev:onpageunload="scwin.onpageunload">
    <xf:group id="grp_search" style="height:31px;">
            <xf:group id="tbl_search" class="w2tb" tagname="table">
                <w2:attributes>
                    <w2:summary></w2:summary>
                </w2:attributes>
                <xf:group tagname="colgroup">
                    <xf:group style="width:15%;" tagname="col"></xf:group>
                    <xf:group style="width:85%;" tagname="col"></xf:group>
                </xf:group>
                <xf:group style="" tagname="tr">
                    <xf:group style="" class="w2tb_th" tagname="th">검색어</xf:group>
                    <xf:group style="" class="w2tb_td" tagname="td">
                        <w2:attributes>
                            <w2:colspan>2</w2:colspan>
                            <w2:rowspan>1</w2:rowspan>
                        </w2:attributes>
                        <xf:input id="ibx_searchKeyword" ref="data:dma_searchParam.SEARCH_KEYWORD" refSync="true"></xf:input>
                        <w2:anchor class="btn_com" id="btn_searchZip" ev:onclick="scwin.btn_searchZip_onclick">
                            <xf:label><![CDATA[검색]]></xf:label>
                        </w2:anchor>
                    </xf:group>
                </xf:group>
            </xf:group>
        </xf:group>
        <w2:gridView dataList="dlt_zipCode" scrollByColumn="false" id="grd_zipCode" useShiftKey="false" autoFit="lastColumn" class="cm_grid">
            <w2:header id="header3" style="">
                <w2:row id="row5" style="">
                    <w2:column blockSelect="false" id="column23" inputType="text" width="85" value="우편번호"></w2:column>
                    <w2:column blockSelect="false" id="column21" width="414" inputType="text" value="주소"></w2:column>
                </w2:row>
            </w2:header>
            <w2:gBody id="gBody3" style="">
                <w2:row id="row6" style="">
                    <w2:column blockSelect="false" id="ZIP_CD" inputType="text" width="85" displayFormat="###-###"></w2:column>
                    <w2:column blockSelect="false" id="ADDRESS" width="414" inputType="text" textAlign="left"></w2:column>
                </w2:row>
            </w2:gBody>
        </w2:gridView>
    </xf:group>
</body>
```

2. 2. 변수 명명 규칙

UI 소스(웹스퀘어 XML) 파일 안에서 사용되는 변수 명명 규칙 정의

명명규칙	• 변수명 허용 문자 : 영문 대소문자, 숫자 • 변수의 첫번째 문자는 영문 소문자로 시작함 • 변수명은 소문자를 기본으로 사용함 • 2개 이상의 단어가 연결되는 경우에는 단어의 첫 글자를 대문자로 작성함 • 변수에 사용되는 단어는 의미 있는 이름을 사용해야 함 • 변수명을 통해서 저장될 데이터의 사용 용도를 추측할 수 있어야 함 • 카멜 표기법에 따름
적용사례	• codeGrpCd, codeGrpNm • billAplySt, billAplyEd
상수선언	• 상수를 선언하려면 영문 대문자와 언더스코어(_) 문자를 구분자로 해서 작성함 • 적용사례 : MAX_COUNT, DEFAULT_TYPE

샘플코드	`// 웹스퀘어 페이지(XML) 내 로컬 변수 선언 예시`
	`scwin.searchZipCodeCallback = function(returnValue) {`
	`  const selectedZipCode = returnValue;  // 로컬 변수 앞에는 반드시 const 또는 let 키워드를 붙여야 함`
	`  adm_memberBasic.set("ZIP_CD", selectedZipCode.ZIP_CD);`
	`};`
	
	`// 웹스퀘어 페이지(XML) 내 전역 변수 선언 예시`
	`scwin.pageItemCnt = 18;`
	
	`scwin.searchFile = function(idx, totalYn) {`
	`  dma_search.set("END_NUM", scwin.pageItemCnt); // 전역 변수는 반드시 scwin 객체 안에 선언해야 함`
	`  $c.sbm.execute(sbm_selectFile);`
	`};`
	
	`// UI 공통 스크립트 내 상수 선언 예시`
	`var gcm = {`
	`  // 서버 통신 기본 미디어 타입 상수 선언`
	`  DEFAULT_OPTIONS_MEDIATYPE : "application/json",`
	`};`
	
	`// 웹스퀘어 페이지(XML) 내 상수 선언 예시`
	`scwin.PAGE_COUNT = 20;`

2. 3. 함수 명명 규칙

UI 소스(웹스퀘어 XML) 파일 안에서 사용되는 함수 명명 규칙 정의

<table>
<tr><td>명명규칙</td><td>• 함수명 허용 문자 : 영문 대소문자
• 함수명 생성 규칙 : 동사(소문자) + 명사형 단어
• 함수명은 소문자를 기본으로 사용함
• 2개 이상의 단어가 연결되는 경우에는 단어의 첫 글자를 대문자로 작성함
• 함수에 사용되는 단어는 의미 있는 이름을 사용해야 함
• 함수명을 통해서 저장될 데이터의 사용 용도를 추측할 수 있어야 함
• 카멜 표기법에 따름</td></tr>
<tr><td>적용사례</td><td>• searchPersonAuthority
• getPersionIAuthority
• saveAuthority
• deleteAuthority</td></tr>
<tr><td>샘플코드</td><td>

```
// UI 공통 자바스크립트 함수 예제
$c.sbm.execute = function(sbmObj, requestData, compObj) {
    if (sbmObj) {
        const sbmObj = (typeof sbmObj == ‘object’) ? sbmObj : (typeof sbmObj == ‘string’) ? $p.getSubmission(sbmObj) : sbmObj;

        if (!sbmObj.action) {
            const alertMsg = $c.data.getMessage(“MSG_CM_00002”, “action”);
            $c.win.alert(alertMsg);
            return false;
        }

        $p.executeSubmission(sbmObj, requestData, compObj);
    }
};

// 웹스퀘어 페이지(XML) 소스 내 함수 작성 예제
scwin.onpageload = function() {
    $c.sbm.execute(sbm_commonSearchItem);
    $c.win.setEnterKeyEvent(tbl_search, scwin.btn_search_onclick);
    $c.util.setGridViewDelCheckBox([grd_groupView, grd_codeView]);
};
```

</td></tr>
</table>

2. 4. 자바스크립트 코딩 규칙

자바스크립트 코드 작성 시 반드시 지켜야 할 코딩 규칙 정의

> 들여쓰기

• 들여쓰기 기본 단위는 Tab으로 설정 (Eclipse Preferences > Text Editors 설정에서 'Display tab width'를 4로 설정하고, 'Insert spaces for tabs' 옵션 체크 해제)

> UI 개발 툴 자동 완성 기능 제공을 위한 문서화 주석 작성

• UI 개발 툴에서 자동 완성 기능(Code Assist) 기능이 필요한 함수의 주석을 작성하기 위해서는 아래와 같은 형식으로 주석을 작성해야 함

```
/**
 * 함수명
 * @date 작성 일자 (ex. 2021.02.26)
 * @memberOf 함수를 선언할 객체 (ex. scwin, $c.win)
 * @param {Object} argument 파라미터 정보
 * @returns {Object} 반환 변수 및 객체
 * @author 작성자명
 * @example 샘플 코드
 */
```

• 주석 작성 샘플

```
/**
 * Alert 메시지 창을 호출한다.
 *
 * @memberof $c.win
 * @date 2020.05.16
 * @param {String} messageStr 메시지
 * @param {String} closeCallbackFncName 콜백 함수명
 * @author Inswave Systems
 * @example
$c.win.alert("우편번호를 선택하시기 바랍니다.");
$c.win.alert("우편번호를 선택하시기 바랍니다.", "scwin.alertCallBack");
*/
com.win.alert = function(messageStr, closeCallbackFncName, opts) {
```

비문서화 주석

- // 와 /* */를 이용해서 주석을 작성함
- 한 줄 주석 작성시에는 // 를 이용해서 주석을 작성함
- 변수 선언 시 변수에 대한 주석을 붙일 경우에는 문장을 마감한 뒤 ';' 뒤쪽에 주석을 작성함

- 주석 작성 샘플

```
scwin.rowcount = 20;  // 레코드열 개수


/**
 * 사원 이미지 파일을 업로드한다.
 */
scwin.btn_uploadImage_onclick = function() {
    // 이미지 경로를 가져온다.
    const rowJSON = dma_MemberBasic.get("IMAGE_PATH");

    // 업로드할 이미지를 선택하고 등록하기 위한 팝업 창을 오픈한다.
    const dataObject = { type : "json",
                         data : { data : rowJSON, callbackFn : "scwin.uploadImageFileCallback" },
                         name : "H3P01" };
    const opt = { id : "H3P01",
                  popupName : "사진 업로드",
                  width : "480", height : "220" };

    /*
     * dataObject : 팝업에 전달할 데이터
     * opt : 팝업 오픈 옵션 설정
     */
    $c.win.openPopup("/ui/HM/HM003P01.xml", opt, dataObject);
};
```

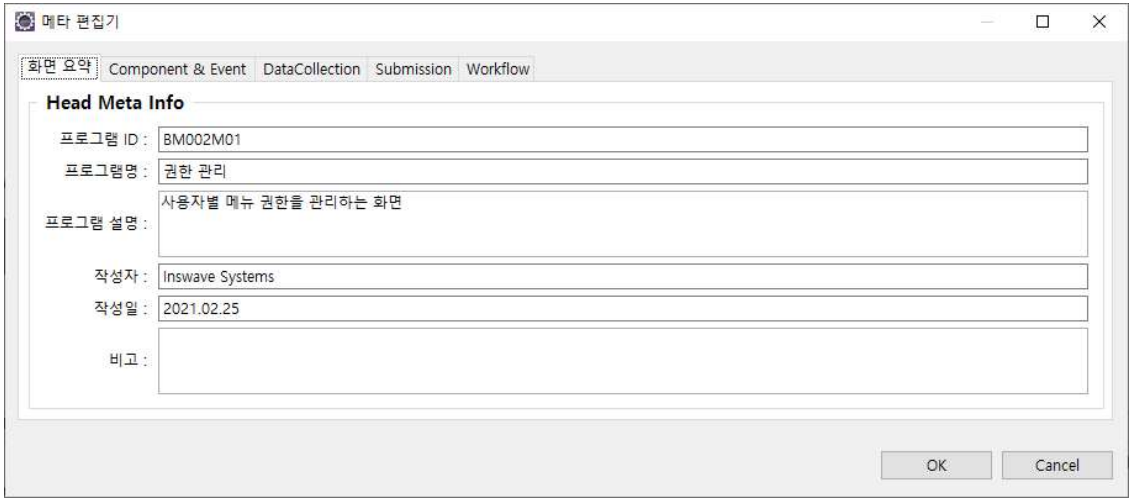
웹스퀘어 페이지(XML) 설명 주석

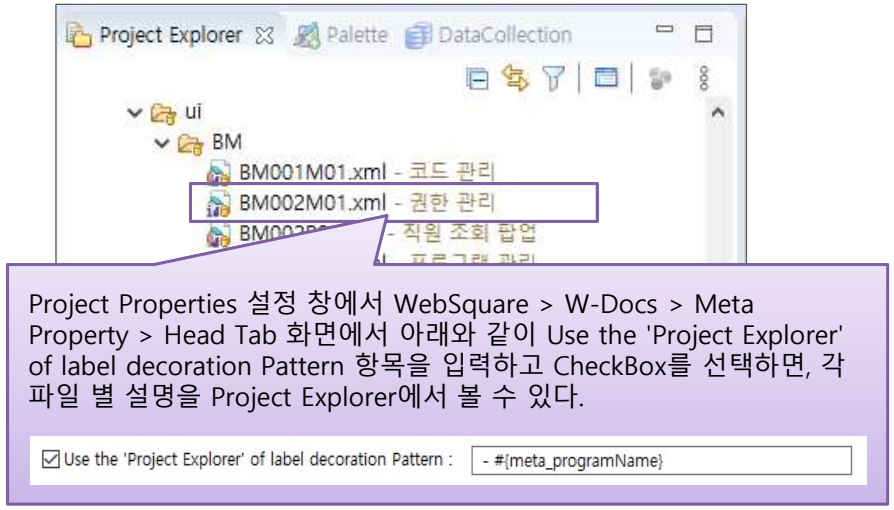
- WebSquare AI Studio 의 메타 편집기를 통해서 웹스퀘어 페이지의 설명 주석을 작성할 수 있다.
- 메타 편집기를 통해서 작성된 주석 샘플

```
<html xmlns="http://www.w3.org/1999/xhtml" xmlns:ev="http://www.w3.org/2001/xml-events" xmlns:w2="http://www.inswave.com/websquare"
    xmlns:xf="http://www.w3.org/2002/xforms">
    <head meta_screenId="BM002M01"
          meta_screenName="권한 관리"
          meta_screenDesc="사용자별 메뉴 권한을 관리하는 화면"
          meta_author="Inswave Systems"
          meta_date="2021.03.19">
        <w2:type>DEFAULT</w2:type>
```

웹스퀘어 페이지(XML) 메타 편집기 오픈 방법

- 웹스퀘어 페이지(XML) 파일 오픈 후, Design View에서 마우스 우클릭 후, [더보기 > 메타 편집기] 메뉴를 선택하고, 아래와 같이 [화면 요약] 탭에서 프로그램 설명을 작성한다. 메타 편집기 화면 오픈 단축키(Ctrl + Shift + Q)를 누르면 바로 화면을 오픈할 수 있다.





함수 선언

- 함수명은 함수 명명 규칙에 맞게 작성해야 함
- 업무 UI 소스의 함수들은 scwin 객체 안에 선언해야 함

- 함수 선언 예제

```
scwin.indexChgSaveCode = function(isReturnVal) {
    if (isReturnVal) {
        scwin.saveData();
    } else {
        const val_grpCdOrg = dlt_commonGrp.getCellData(scwin.idxRow, "GRP_CD_ORG");
        if (val_grpCdOrg) {
            dma_commonGrp.set("GRP_CD", val_grpCdOrg);
            $c.sbm.execute(sbm_commonCode);
        }
    }
};
```

전역 변수 선언

- 전역 변수명은 변수 명명 규칙에 맞게 작성해야 함
- 업무 UI 소스 내의 전역 변수들은 scwin 객체 안에 선언해야 함

- 전역 변수 선언 예제

```
/**
 * 마지막에 선택한 Row Index (전역 변수)
 */
scwin.idxRow = 0;

scwin.grd_groupView_onrowindexchange = function(row, oldRow) {
    scwin.idxRow = row;
}
```

로컬 변수 선언

- 로컬 변수명은 변수 명명 규칙에 맞게 작성해야 함
- 함수 내에서 사용하는 로컬 변수들은 변수명 앞에 let 또는 const를 선언해야 함 (변수 선언 없는 경우 window 객체 아래 생성되는 전역 변수로 생성됨)

- 로컬 변수 선언 예제

```
scwin.btn_modify_onclick = function() {
    // selectedData, data, options 변수는 함수 내에서만 존재하고 접근이 가능한 로컬 변수임
    const selectedData = dlt_fileGroup.getRowJSON(dlt_fileGroup.getRowPosition());
    const data = {data : selectedData, mode : "update", callbackFn : "scwin.modCallback" };
    const options = {
        id : "FilegroupPop",
        popupName : "파일 그룹 설정",
        modal : true,
        width : 600, height: 344 };
    $c.win.openPopup("/ui/BM/BM006P01.xml", options, data);
};
```

상수 선언

- 상수명은 영문 대문자, 언더스코어(_)를 사용해서 작성해야 함
- 업무 UI 소스 내의 상수는 scwin 객체 안에 선언해야 함

- 상수 선언 예제

```
scwin.MAX_REGISTER_COUNT = 30;
```

문장 작성

- 한 줄에 하나의 문장을 써야 함
- 중첩된 문장에서는 상위 문장의 시작에서 들여 쓰기를 해야 함

- 올바른 작성 예제

```
scwin.btn_search_onclick = function() {
    if (sbx_choiceBox.getValue() == "MENU_LEVEL") {
        dma_search.set("CONTENT", sbx_menuLevel.getValue());
    } else {
        dma_search.set("CONTENT", acb_autoChoiceBox.getValue());
    }
};
```

- 잘못된 작성 예제

```
scwin.btn_search_onclick = function() {
if (sbx_choiceBox.getValue() == "MENU_LEVEL") {
    dma_search.set("CONTENT", sbx_menuLevel.getValue());
    } else {
    dma_search.set("CONTENT", acb_autoChoiceBox.getValue());
    }
};
```

if - else 문 작성

- 반드시 내용을 중괄호로 묶어야 함. if 와 '(' 사이에 공백을 하나 주어야 함
- ')' 뒤에 시작 중괄호가 '{' 옴.)'와 '{' 사이에 공백을 하나 주어야 함

```
scwin.btn_search_onclick = function() {
    if (sbx_choiceBox.getValue() == "MENU_LEVEL") {
        dma_search.set("CONTENT", sbx_menuLevel.getValue());
    } else {
        dma_search.set("CONTENT", acb_autoChoiceBox.getValue());
    }
};
```

for 문 작성

- 반드시 내용을 중괄호로 묶어야 함. for 와 '(' 사이에 공백을 하나 주어야 함
- ';' 뒤에 하나의 공백을 주어야 함. ')'와 '{' 사이에 하나의 공백을 주어야 함
- Iteration 변수는 i, j, k를 사용하지 말고 idx1, idx2, idx3을 사용해야 함
- for 문 작성 예제

```
for (let idx1 = 0; i < jsonData.length; idx1++) {
    for (let idx2 = 0; i < dataLength1; idx2++) {
        jsonData[idx1][idx2] = idx1 * idx2;
    }
}
```

while 문 작성

- while 와 '(' 사이에 공백을 하나 주어야 함
- ')'와 '{' 사이에 하나의 공백을 주어야 함
- while 문 작성 예제

```
let idx = 1;
while ((idx <= 30) && (status === false)) {
    data[idx] = idx;
    idx++;
}
```

do - while 문 작성

- do, while 와 '(' 사이에 공백을 하나 주어야 함
- '}' 와 while 사이에 공백을 하나 주어야 함
- do - while 문 작성 예제

```
let idx = 1;
do {
    data[idx] = idx;
    idx++;
} while ((idx <= 30) && (status === false));
```

switch문 작성

- switch와 '(' 사이에 공백을 하나 주어야 함
- ')' 와 '{' 사이에 공백을 하나 주어야 함
- case는 tab 하나만큼 들여서 시작해야 함
- 각 case 시작전에 공백 한 줄을 주어야 함

- switch 문 작성 예제

```
switch (tab_content.getSelectedTabID()) {
    case 1:
        scwin.createChartData();
        break;
    case 2:
        scwin.selectPrdyPrgrs();
        break;
    case 3:
        scwin.selectInvstr();
        break;
    case 4:
        scwin.selectItemsNews();
        break;
}
```

try - catch문 작성

- try 와 '{' 사이에 공백을 하나 주어야 함
- '}' 와 catch 사이에 공백을 하나 주어야 함
- catch 와 '(' 사이에 공백을 하나 주어야 함
- ')' 와 '{' 사이에 공백을 하나 주어야 함
- '}' 와 finally 사이에 공백을 하나 주어야 함
- finally와 '{' 사이에 공백을 하나 주어야 함

- try - catch 문 작성 예제

```
$c.removeGridView = function(gridViewObj) {
    try {
        if ((typeof gridViewObj === "object") && (gridViewObj !== "")) {

            const dltObj = this.getGridViewDataList(gridViewObj);

            if (dltObj === null) {
                return;
            }

            const checkedIdxArr = dltObj.getMatchedIndex("CHK", "1", true, 0, dltObj.getRowCount());

            if (checkedIdxArr.length > 0) {
                for (let i = (checkedIdxArr.length - 1); i >= 0 ; i--) {
                    dltObj.removeRow(checkedIdxArr[i]);
                }
            } else {
                const focusedRowIdx = gridViewObj.getFocusedRowIndex();
                dltObj.removeRow(checkedIdxArr[i]);
            }
        }
    } catch(e) {
        $p.log("[com.removeGridView] Exception :: " + e.message);
    } finally {
        dltObj = null;
    }
};
```