

WRM(WebSquare Reference Model) for WebSquare AI

WRM W-Pack 적용 가이드

2023.07.28

재개정 이력

문서 버전	작성일자	작성자	설명
1.0	2023-07-28	박상규	초안 작성
1.1	2024-05-22	이해룡	WRM Websquare AI 적용

Table of Contents

1. W-Pack을 이용한 웹스퀘어 페이지 및 JS 파일 내 코드 \$p 객체 전달 기능 적용
2. W-Pack 공통 함수 Scope 전달 관련 옵션 설정 및 빌드
3. W-Pack scope 관련 config.xml 옵션 설정 내역
4. config.xml SPA 옵션에 따른 com 객체 생성 방식
5. W-Pack scope 옵션을 위한 WRM 코드 수정
6. WebSquare5 SP5 Studio 개선 사항
7. WebSquare5 SP5 Engine 개선 사항

■ 공통 함수 객체 clone 기능 개선

- 현재의 clone 기능은 wframe 개수만큼 공통 객체를 복제하고 있다. wframe이 적은 경우에는 문제가 되지 않으나 wframe이 많을 경우 성능에 큰 문제가 될 수 있다.
- 특히 UDC 2.0을 사용하는 경우 UDC 개수만큼 공통 함수 객체를 복제하기 때문에, 현재의 clone 기능이 화면 로딩 속도 저하 및 메모리 사용량 증가, 화면 종료 시 화면 Clear 처리 속도 저하를 가져올 수 있다.
- clone 기능은 현재 config에서 <spa>의 <variable clone="com">과 같이 설정한다.
- wframe에서 공통 객체를 복제하지 않고도 공통 함수 객체의 정의된 함수 호출 시, \$p 파라미터를 전달할 수 있는 기능을 W-Pack을 통해 제공한다.
- W-Pack으로 빌드 시 공통 함수 객체(com)내 함수를 호출하는 코드에 \$p를 첫번째 파라미터에 자동으로 추가해줌으로써, clone 기능을 사용하지 않고도, 공통 함수 안에서 자신을 호출한 WFrame의 Scope을 찾을 수 있도록 한다.

■ W-Pack 공통 함수 호출 코드 변환 규칙

- 공통 함수 객체를 복제하는 대신 공통 함수의 선언부 및 호출부를 W-Pack을 통해 변환하여 Scope내 \$p 객체를 전달해서 공통 함수 객체를 복제하지 않고도 Scope을 인식해서 동작할 수 있도록 한다.

■ 변환 대상

- W-Pack으로 전환되는 웹스퀘어 페이지 파일(XML)
- config.xml 파일에 [wpack > commonPath]에 등록한 공통 함수 파일(JS)

■ 변환 규칙

- 공통 함수 선언부를 찾아 첫번째 파라미터로 \$p 객체 추가

`$c.win.openPopup = function(url, opts, data) { } → $c.win.openPopup = function($p, url, opts, data) { }`

- 공통 함수 호출부를 찾아 첫번째 파라미터로 \$p 객체 추가

`$c.win.openPopup(url, opts, data) → $c.win.openPopup($p, url, opts, data);`

2. W-Pack 공통 함수 Scope 전달 관련 옵션 설정 및 빌드

■ W-Pack을 이용해서 공통 함수 객체 내 함수에 scope 전달하기 위해서 \$p 파라미터를 전달하기 위한 추가 옵션

Option	Description
-sc, --scopeCommon	<ul style="list-style-type: none"> Scope 전달을 위해서 첫번째 파라미터로 \$p 객체를 자동으로 추가할 공통 함수 객체 설정 공통 함수 객체가 여러 개인 경우 구분자(.)를 통해 나열해서 선언함 <p>※ 옵션 설정 예제</p> <pre>--scopeCommon '\$c'</pre> <pre>--scopeCommon '\$c, biz' // 공통 함수 객체가 2개인 경우(\$c과 biz)</pre> <pre>-sc '\$c'</pre>
-gp, --globalPath	<ul style="list-style-type: none"> Scope 전달을 위해서 첫번째 파라미터로 \$p 객체를 자동으로 추가할 공통 함수를 작성한 JS 파일 경로 및 파일명 설정 W-Pack으로 빌드하면 공통함수 선언 및 호출 시 맨 앞에 \$p 파라미터를 자동으로 추가함 commonPath 경로에 있는 js 파일들은 scopeCommon으로 지정한 공통 함수 객체를 대상으로 변환하여 dest 디렉터리에 생성된 JS 파일이 저장됨 개별 파일 단위 또는 디렉터리 단위로 설정할 수 있음 디렉터리로 설정할 경우 디렉터리 안에 모든 js 파일을 W-Pack으로 빌드함 한 디렉터리 안에 W-Pack으로 빌드할 js 파일과 빌드 대상이 아닌 js 파일이 혼재될 경우, W-Pack으로 빌드할 JS 파일과 빌드하지 않을 JS 파일을 분리해서 2개의 디렉터리로 구분해서 저장하거나, W-Pack으로 빌드할 파일만 각 파일 별로 옵션 설정을 해야 함 <p>※ 옵션 설정 예제</p> <pre><예제1> --globalPath "/cm/gcc/data.xml"</pre> <pre><예제2> -gp</pre> <pre>"/cm/gcc/data.xml,/cm/gcc/sbm.xml,/cm/gcc/win.xml,/cm/gcc/util.xml,/cm/gcc/date.xml,/cm/gcc/num.xml,/cm/gcc/str.xml,/cm/gcc/hkey.xml,/cm/gcc/ext.xml"</pre> <pre><예제3> --dest "E:/Eclipse_Workspace/GitHub_workspace/DEV_PACK/WRM/WebContent/_wpack_" -gp "/cm/gcc/data.xml"</pre>

2. W-Pack scope 관련 옵션 설정 및 빌드

■ scope 관련 옵션을 추가한 WRM W-Pack 빌드 명령어

- WRM 전체 소스 빌드를 위한 W-Pack 실행 배치 파일 위치 : C:\WEBSQUARE_DEV_PACK_SP5\bin\w-pack\standalone_wpack-win.bat

```
C:/WEBSQUARE_DEV_PACK_AI/bin/w-pack/standalone_wpack-win.exe --baseDir "C:/WEBSQUARE_DEV_PACK_AI/workspace/WRM/WebContent" --src "C:/WEBSQUARE_DEV_PACK_AI/workspace/WRM/WebContent" --dest "C:/WEBSQUARE_DEV_PACK_AI/workspace/WRM/WebContent/_wpack_" --js "0" --css "0" -sc '$c'
```

Option	Description
--baseDir "C:/WEBSQUARE_DEV_PACK_AI/workspace/WRM/WebContent"	<ul style="list-style-type: none">wpack으로 빌드하려는 WebContent 기준(Root) 경로
--src "C:/WEBSQUARE_DEV_PACK_AI/workspace/WRM/WebContent"	<ul style="list-style-type: none">wpack으로 빌드하려는 xml, js 파일이 저장된 경로
--dest "C:/WEBSQUARE_DEV_PACK_AI/workspace/WRM/WebContent/_wpack_"	<ul style="list-style-type: none">--src에 정의된 xml, js 파일을 wpack으로 빌드한 js 파일을 저장할 경로
--js "0"	<ul style="list-style-type: none">0 : javascript 코드를 minify 하지 않음1 : javascript 코드를 minify하며 주석 및 공백을 제거함2 : javascript 코드를 minify하며 주석 및 공백 제거, 로컬 변수를 치환함
--css "0"	<ul style="list-style-type: none">0 : css 코드를 minify 하지 않음1 : css 코드 주석 및 공백 제거, 색상 값 (RGB -> HEX) 변환 처리함
-sc '\$c'	<ul style="list-style-type: none">공통 함수 객체(\$c)에 \$p 파라미터를 전달하도록 함수 코드 생성
-cp "/cm/js"	<ul style="list-style-type: none">공통 함수 객체(com)가 선언된 js 파일 경로 설정 (디렉터리 및 파일 단위 설정 가능)디렉터리로 설정할 경우 디렉터리 안에 모든 js 파일을 W-Pack으로 빌드함디렉터리 안에 W-Pack으로 빌드할 js 파일과 빌드 대상이 아닌 js 파일이 혼재될 경우, 디렉터리를 분리해서 js 파일을 저장하거나 개별 파일 단위로 옵션 설정을 해야 함<예제1> - cp "/cm/js/commonGlobal.js, /cm/js/commonScope.js"<예제2> - cp "/cm/js"

2. W-Pack scope 관련 옵션 설정 및 빌드

■ 전체 W-Pack 옵션 정보

Option	Description
-V, --version	<ul style="list-style-type: none">W-Pack 버전 정보
-m, --module <type>	<ul style="list-style-type: none">변환 타입 --module W-Pack/sass/reverse (default: W-Pack)
-b, --baseDir <path>	<ul style="list-style-type: none">project의 경로 --baseDir /Users/kims/project/
-s, --src <path>	<ul style="list-style-type: none">input 디렉토리 경로 --src ./src/xml/ (필수)
-d, --dest <path>	<ul style="list-style-type: none">output 디렉토리 경로 --dest ./src/js/ (필수)
-j, --js <level>	<ul style="list-style-type: none">JS Minify 레벨<ul style="list-style-type: none">0 : javascript 코드를 minify 하지 않음1 : javascript 코드를 minify하며 주석 및 공백을 제거함2 : javascript 코드를 minify하며 주석 및 공백 제거, 로컬 변수를 치환함
-c, --css <level>	<ul style="list-style-type: none">CSS Minify 레벨<ul style="list-style-type: none">0 : css 코드를 minify 하지 않음1 : css 코드 주석 및 공백 제거, 색상 값 (RGB -> HEX) 변환 처리함
-i, --ignore <path>	<ul style="list-style-type: none">-src에서 설정한 디렉터리 내 하위 디렉터리에서 파일을 W-Pack으로 빌드 시 제외시킬 디렉터리 경로
-nd, --noDebug <boolean>	<ul style="list-style-type: none">minify중에 debugger코드를 남기려면 false, 기본은 true
-ct, --context <path>	<ul style="list-style-type: none">sourceMap 생성시 사용할 context경로

2. W-Pack scope 관련 옵션 설정 및 빌드

■ 전체 W-Pack 옵션 정보

Option	Description
-bb, --babel <boolean>	<ul style="list-style-type: none">babel 사용 여부
-l, --loader <boolean>	<ul style="list-style-type: none">[import, requirejs] module loader 설정. 기본값은 import(ES6 브라우저 기능인 import).
-ns, --noSourceMap <boolean>	<ul style="list-style-type: none">minify시 sourcemap을 생성하지 않는 경우엔 true, 기본은 false
-nb, --noFixBaseDir <boolean>	<ul style="list-style-type: none">baseDir이 dest의 일부분이 아닌것을 허용할때 true, 기본은 false
-pm, --parseMeta <boolean>	<ul style="list-style-type: none">meta 관련 정보를 parsing하려면 true, 기본값은 false
-sc, --scopeCommon <string>	<ul style="list-style-type: none">Scope 전달을 위해서 첫번째 파라미터로 \$p 객체를 자동으로 추가할 공통 함수 객체 설정공통 함수 객체가 여러 개인 경우 구분자(.)를 통해 나열해서 선언함
-sn, --scopeName <string>	<ul style="list-style-type: none">\$p 대신 다른 이름을 사용하는 경우 사용한다. 기본값은 \$p
-cp, --commonPath <path>	<ul style="list-style-type: none">Scope 전달을 위해서 첫번째 파라미터로 \$p 객체를 자동으로 추가할 공통 객체 및 함수를 선언한 JS 파일 경로 및 이름 설정W-Pack으로 빌드하면 공통함수 선언 및 호출 시 맨 앞에 \$p 파라미터를 자동으로 추가함commonPath 경로에 있는 js 파일들은 scopeCommon으로 지정한 변수들을 치환하여 dest 폴더에 변환된 js가 생성됨개별 파일 단위 또는 디렉터리 단위로 설정할 수 있음디렉터리로 설정할 경우 디렉터리 안에 모든 js 파일을 W-Pack으로 빌드함디렉터리 안에 W-Pack으로 빌드할 js 파일과 빌드 대상이 아닌 js 파일이 혼재될 경우, 디렉터리를 분리해서 js 파일을 저장하거나 개별 파일 단위로 옵션 설정을 해야 함
-h, --help	<ul style="list-style-type: none">W-Pack 명령어 도움말 표시

2. W-Pack scope 관련 옵션 설정 및 빌드

■ 전체 W-Pack 옵션 정보

Option	Description
-gp, --globalPath <string>	<ul style="list-style-type: none">• 치환 기능을 사용할 global common component 경로.• 해당 경로에 있는 xml 파일들은 scwin으로 정의된 함수에 \$p가 첫번째 인자로 추가됨
-gc, --globalCommon <string>	<ul style="list-style-type: none">• lobaPath 기능 사용 시 치환 대상이 되는 scope 변수명. 기본값은 scwin
-ep, --excludePrefix <boolean>	<ul style="list-style-type: none">• 특정 함수에 치환 기능을 적용하지 않을 경우에 사용. 이 옵션 이름으로 시작하는 함수명은 치환 기능이 미적용

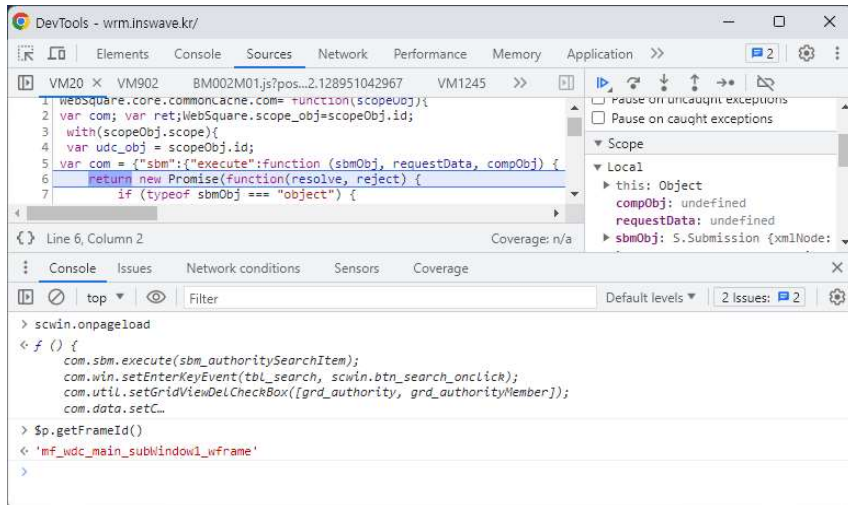
3. W-Pack scope 관련 config.xml 옵션 설정 내역

```
<WebSquare>
  <wpack use="true">
    <contextRoot value="/" />
    <srcExtension value="xml" />
    <destExtension value="js" />
    <destRoot value="_wpack_" />
    <babelDestRoot value="_wpackbabel_" />
    <loadingMode value="auto" />
    <common name="" value="false" />
    <commonPath/>
  </wpack>
  <spa>
    <!-- clone="com" 옵션 설정이 삭제됨 -->
    <variable value="scwin" />
    <scriptCache value="true" />
    <autoReload count="50" value="true" />
  </spa>
</WebSquare>
```

4. config.xml SPA 옵션에 따른 com 객체 생성 방식

■ TYPE – 1 (WRM 5.0.5.5 이하 버전에 적용한 방식)

```
<WebSquare>
  <spa>
    <variable value="scwin" clone="com" />
  </spa>
</engine>
<module src="/cm/js/commonGlobal.js" />
<module src="/cm/js/commonScope.js" />
</engine>
</WebSquare>
```

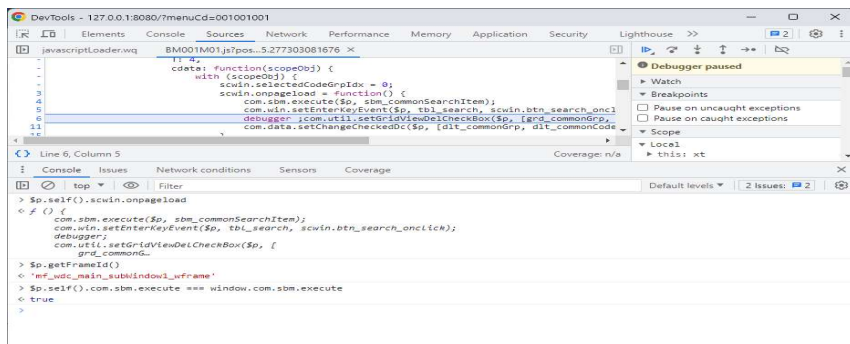


- WFrame Scope 별로 공통 객체(com)을 복제(Clone)하는 방식
- WFrame Scope 별로 공통 함수 객체(com) 함수를 복제하고 있음
- \$p, 공통 객체(com), 웹스퀘어 페이지(XML) 내 객체(scwin) 모두 공통 객체(com) 내 함수 내에서도 공통 객체(com) 내 함수를 호출한 웹스퀘어 페이지(XML)와 동일한 Scope을 유지하고 실행됨 (따라서, 공통 함수 내에서 \$p.getFrameId(), scwin.onpageLoad(), com.sbm.execute(sbm_search1)를 호출해도 해당 함수를 호출한 웹스퀘어 페이지 내의 스크립트와 동일한 Scope을 유지하며 실행됨
- 전체 객체 window.com 객체를 Scope별로 com 객체(ex. mf_wdc_main_subWindow1_wframe_com)를 생성해서 복제할 때 com 객체 내 변수와 함수 모두 복제를 함
- Scope 내 com과 window.com 내 함수를 비교하면 아래와 같은 false 가 리턴되는 것을 확인할 수 있음 (ex. com.data.createDataList === window.com.data.createDataList >>> false)
- 변수와 함수 복제를 모두 하기 때문에 공통 함수 객체(com)내 선언한 함수 개수가 많을 수록 메모리 사용량이 크게 늘어날 수 있음

4. config.xml SPA 옵션에 따른 com 객체 생성 방식

■ TYPE - 2

```
<WebSquare>
  <wpack>
    <common value="true" name="com" />
    <commonPath>
      <path src="/cm/js/commonGlobal.js"></path>
      <path src="/cm/js/commonScope.js"></path>
      <path src="/cm/js/main.js"></path>
    </commonPath>
  </wpack>
  <spa>
    <variable value="scwin" link="com" />
  </spa>
  <engine>
    <module src="/cm/js/commonGlobal.js" />
    <module src="/cm/js/commonScope.js" />
  </engine>
</WebSquare>
```



- WFrame Scope 별로 공통 객체(com)을 링크(Link)하는 방식
- WFrame Scope 별로 공통 함수 객체(com) 함수를 생성은 하나 함수 코드를 복제하지 않고 전역 공통 함수 객체(window.com) 객체에 링크를 걸고 있음
(ex. mf_wdc_main_subWindow0_wframe_com.data.createDataList = window.com.data.createDataList)
- Scope 내 com과 window.com 내 함수를 비교하면 아래와 같은 true 가 리턴되는 것을 확인할 수 있음 (ex. com.data.createDataList === window.com.data.createDataList >>> true)
- wpack에서 원본 JS 을 JS로 빌드 시 \$p 객체로 공통 함수 파라미터로 전달하기 때문에, \$p 객체는 함수는 복제(Clone) 시와 동일하게 사용할 수 있으며, com 객체의 변수를 선언해서 사용한 경우에는 앞에 "\$p.self()."을 붙여서 치환해 주고 있음 (ex. com.win.popup = {} → \$p.self().com.win.popup = {})
- 공통 함수 객체에서는 웹스퀘어 페이지(XML) 내 객체(scwin)를 직접 참조해서는 안됨
- 전체 객체 window.com 객체를 Scope별로 com 객체(ex. mf_wdc_main_subWindow1_wframe_com)를 생성해서 복제할 때 com 객체 내 변수만 복제하고, 함수는 window.com 객체 내 동일한 함수와 링크를 걸고 있음
- 변수만 복제하고 함수는 링크만 걸기 때문에 com 객체 내 함수 개수가 많아도 메모리 사용량이 복제(clone) 방식보다 적게 늘어남
- WFrame Scope 단위로 생성한 com 객체를 제거하기 힘든 프로젝트에서 코드 수정을 최소화하면서 Clone 기능 사용으로 인한 성능 저하 개선 효과를 얻을 수 있음

5. W-Pack scope 옵션을 위한 WRM 코드 수정

■ WRM에서 wpack scope 옵션 적용을 위해서 공통 함수 수정

- commonGlobal.js 파일에서 gcm.win.openPopup() 함수 내 아래의 코드를 수정함
- com 객체 대신에 gcm 객체에 정의된 gcm.data.getParameter 함수를 호출하고, \$p 객체를 직접 전달해 줌

```
// var com = popupWindow.com;
// var messageType = com.data.getParameter("messageType") || "alert";
// var callbackFuncStr = com.data.getParameter("callbackFn");
var messageType = $c.data.getParameter(popupWindow.$p, "messageType") || "alert";
Var callbackFuncStr = $c.data.getParameter(popupWindow.$p, "callbackFn");
```

- commonScope.js 파일에서 com.win.closePopup 함수 내 아래의 코드를 수정함
- com 객체 대신에 scwin 객체에 콜백 함수에 전달할 파라미터 객체를 저장하도록 수정함

```
$c.win.closePopup = function(callbackParam, callbackFnStr) {
    if (!$c.util.isEmpty(callbackParam)) {
        // com.win.popup = {
        //     callbackParam : callbackParam
        // };
        $p.getFrame().scope.scwin._popup = {
            callbackParam : callbackParam
        };
    }
};
```

5. W-Pack scope 옵션을 위한 WRM 코드 수정

■ W-Pack으로 빌드된 sbm.js 파일 내 \$c.sbm.executeDynamic 함수 코드

```
$c.sbm.executeDynamic = function ($p, options, requestData, obj) {  
    var submissionObj = $c.util.getComponent($p, options.id);  
    if (submissionObj === null) {  
        $c.sbm.create($p, options);  
        submissionObj = $c.util.getComponent($p, options.id);  
    } else {  
        $p.deleteSubmission(options.id);  
        $c.sbm.create($p, options);  
        submissionObj = $c.util.getComponent($p, options.id);  
    }  
    return $c.sbm.execute($p, submissionObj, requestData, obj);  
};
```

■ BM001M01.xml 파일을 W-Pack으로 빌드한 BM001M01.js 파일 내 scwin.onpageload 함수 코드

```
scwin.onpageload = function () {  
    $c.sbm.execute($p, sbm_commonSearchItem);  
    $c.win.setEnterKeyEvent($p, tbl_search, scwin.btn_search_onclick);  
    $c.util.setGridViewDelCheckBox($p, [  
        grd_commonGrp,  
        grd_commonCode  
    ]);  
    $c.data.setChangeCheckedDc($p, [  
        dlt_commonGrp,  
        dlt_commonCode  
    ]);  
};
```

5. W-Pack scope 옵션을 위한 WRM 코드 수정

■ WRM에서 config.xml 파일의 initScript, postScript

- config.xml 파일의 initScript와 postScript 태그에 작성한 코드는 wpack 빌드를 하지 않기 때문에 직접 \$p 객체를 전달하도록 수동으로 코드를 수정해야 함

```
<initScript value="true"><![CDATA[
    $c.hkey.init($p);
    $c.hkey.setShortKey($p, $p.main());
    $c.win.setLangCode($p, $c.win.getLanguage($p));
    $c.win.setHistoryBackEvent();
]]></initScript>
<postScript value="false"/>
<wframe>
    <scope value="true"/>
    <mode value="async"/>
    <initScript value="true"><![CDATA[
        $c.hkey.setShortKey($p);
        $c.win.setProgramAuthority($p);
        $c.data.initChangeCheckedDc($p);
    ]]></initScript>
    <postScript value="true"><![CDATA[
        $c.win.processCommonData($p);
    ]]></postScript>
    <popupAutoClose value="true"/>
</wframe>
```

6. WebSquare AI Studio 개선 사항

WRM W-Pack 적용 가이드

client.config.xml ×

Common 엔진 초기 설정으로 복원

Project UDC

Project UDC :

name	src
\$c.data	/cm/gcc/data.xml
\$c.sbm	/cm/gcc/sbm.xml
\$c.win	/cm/gcc/win.xml
\$c.util	/cm/gcc/util.xml
\$c.date	/cm/gcc/date.xml
\$c.num	/cm/gcc/num.xml
\$c.str	/cm/gcc/str.xml
\$c.hkey	/cm/gcc/hkey.xml
\$c.ext	/cm/gcc/ext.xml

Add... Edit Remove

공통 모듈로 사용되는 Project UDC를 등록합니다.
* name : Project UDC에 접근할 때 사용하는 변수명
* src : Project UDC의 경로

config.xml 파일에 [common> Project UDC] 속성 설정이 추가됨

6. WebSquare AI Studio 개선 사항

WRM W-Pack 적용 가이드

Project Explorer

- Workspace
- Deployed Resources
- target
- WebContent
- _wpack_
- cm
- gcc
- data.js
- date.js
- ext.js
- hkey.js
- num.js
- sbm.js
- str.js
- util.js
- win.js
- main
- t
- ui
- WE
- web
- fileUp
- ui
- BM
- HM
- SP
- WEB-INF
- websquare

client.config.xml

Common

Project UDC

Project UDC :	name	src
	\$c.date	/cm/gcc/date.xml
	\$c.num	/cm/gcc/num.xml
	\$c.str	/cm/gcc/str.xml
	\$c.hkey	/cm/gcc/hkey.xml
	\$c.ext	/cm/gcc/ext.xml

Add... Edit Remove

공통 모듈로 사용되는 Project UDC를 등록합니다.
* name : Project UDC에 접근할 때 사용하는 변수명
* src : Project UDC의 경로

MSA

MSA server :

name	origin	w2xHome	baseURI	baseServletURI	contextRoot
------	--------	---------	---------	----------------	-------------

Add... Edit Remove

MSA 서비스 단위인 서버 이름과 url을 설정합니다.
* name : 등록된 MSA server의 이름입니다.
* origin : 등록된 MSA server의 origin 경로입니다.
(ex : https://www.inswave.com:9090)

공통 모듈로 사용되는 MSA UDC를 등록합니다.
* MSA server : MSA UDC가 위치한 MSA server 이름
* name : MSA UDC에 접근할 때 사용하는 변수명

MSA Client LanguagePack :

msaName	lang	value
---------	------	-------

Add... Edit Remove

MSA Client 다국어 json 파일의 경로를 등록합니다.
* msaName : MSA 서버 이름
* lang : 언어 지정 (ex) ko, en

config.xml 파일의 [common > Project UDC] 속성에 정의된 경로의 JS 파일이 수정되면, WebSquare AI Studio에서 JS 파일을 빌드해서 _wpack_ 경로 하위에 생성된 파일을 저장함

7. WebSquare AI Engine 개선 사항

WRM W-Pack 적용 가이드

The screenshot shows the inswave WebSquare AI Engine interface. A callout box highlights the configuration for including W-Pack files in the XML file:

config.xml 파일의 [common > Project UDC] 속성에 정의된 경로의 xml 파일을 웹스퀘어 페이지(XML) 파일에서 script 태그로 Include하면, "_wpack_" 경로 하위에 W-Pack으로 빌드된 JS 파일을 요청하도록 웹스퀘어 AI엔진에 기능이 추가됨

The network inspector on the right shows the following request headers for the fileMultiUpload.js request:

- Request URL: http://localhost:8080/_wpack/_cm/gcc/data.js?postfix=1716423405384113372548
- Request Method: GET
- Status Code: 200 OK
- Remote Address: [::1]:8080
- Referrer Policy: strict-origin-when-cross-origin

The response headers show the following details:

- Accept-Ranges: bytes
- Content-Length: 118930
- Content-Type: application/javascript
- Date: Thu, 23 May 2024 05:32:20 GMT
- Last-Modified: Mon, 20 May 2024 16:20:51 GMT

The request headers show the following details:

- Accept: */*
- Accept-Encoding: gzip, deflate, br, zstd
- Accept-Language: ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
- Cache-Control: no-cache
- Connection: keep-alive
- Cookie: _USER_ID_SAVE_YN=; _LOGIN_USER_ID=; system_language=ko; lastAccess=1716364293304; JSESSIONID=3003E0EA7F9E4EDC25757F7E2CCDB664; globalDebug=false
- Host: localhost:8080
- Pragma: no-cache
- Referer: http://localhost:8080/
- Sec-Ch-Ua: "Google Chrome";v="125", "Chromium";v="125", "Not.A/Brand";v="24"
- Sec-Ch-Ua-Mobile: ?0

Thank You

감사합니다.

홈페이지	http://www.inswave.com
웹스퀘어	http://www.websquare.co.kr
기술지원	http://wtech.inswave.kr

Beyond Web Technology, Create Our Business Value

웹을 넘어, 새로운 비즈니스 가치의 창출

inswave systems