

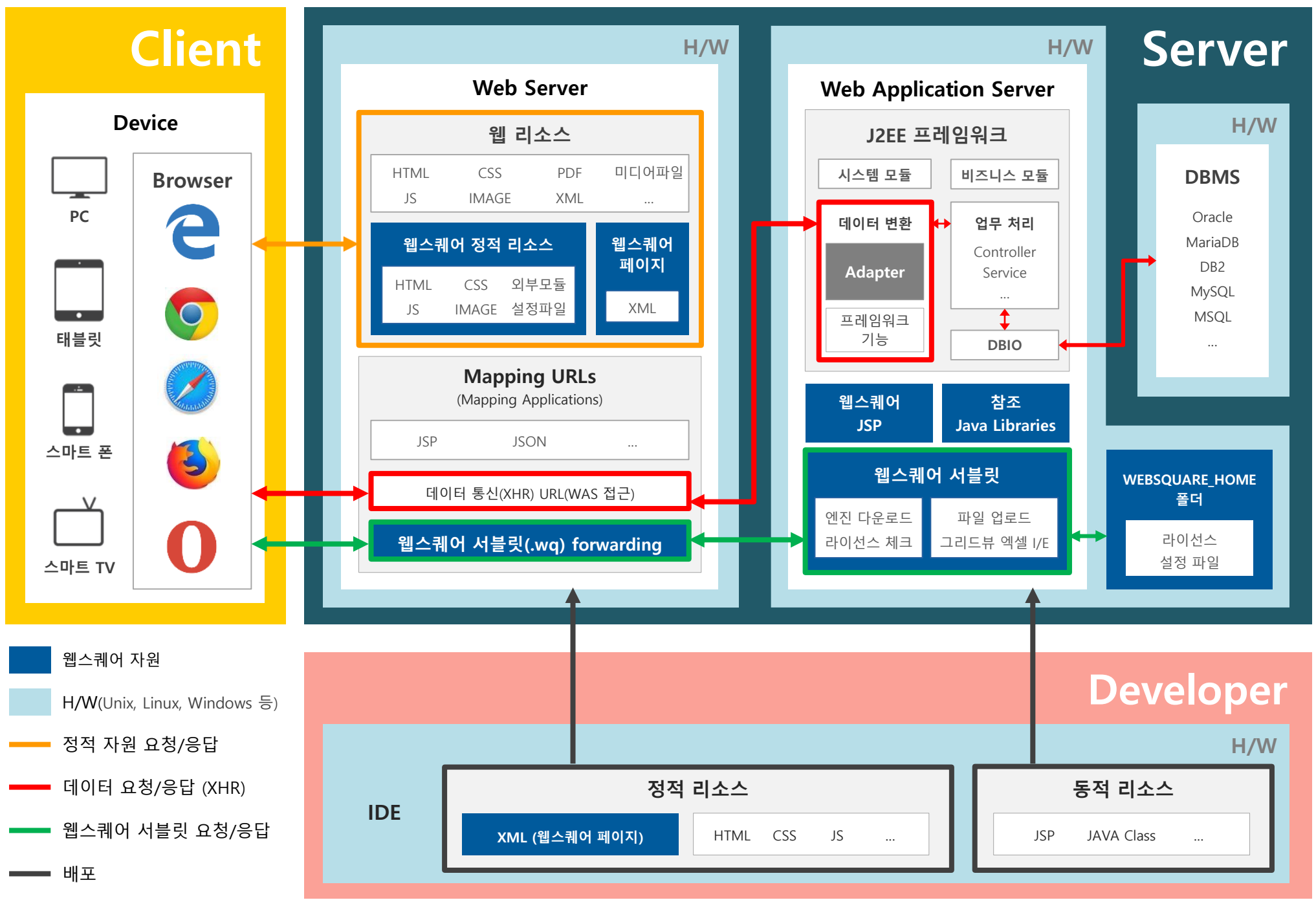
WebSquare5 심화

이 과정의 목적은 아래와 같습니다.

- 1) WebSquare5로 빠르고 가벼운 웹 페이지 구현의 가이드 라인을 제시하기 위함에 있으며, 이를 위해 페이지 구현을 할 때 준수하고 고려해야 하는 사항들을 소개합니다.
- 2) WebSquare5의 컴포넌트들을 이용하여 구현한 확장 샘플들을 소개합니다.

- 웹의 성능 최적화
- WebSquare5 SPA
- Page Scope Coding
- 성능 향상을 위한 웹스퀘어 API
- 웹스퀘어 컴포넌트 활용

| 웹의 성능 최적화



✓ lazy='false' Mode

※ **WebSquare SP1(2015년 11월 이후 엔진)** 부터 아래와 같은 ScriptMode가 추가되면서

onpageload 및 onpageunload 이벤트가 자동 생성됩니다.

해당 내용이 적용된 이유는 웹스퀘어5에서 사용하는 일부 formatter 및 callback 함수의 동작 시점을 보정하기 위함입니다.

(예: customFormatter, invalidMessageFunc, callback 함수 등)

■ lazy="false" 설정 및 onpageload 및 onpageunload 이벤트 추가

[AS-IS]

```
<script type="text/javascript">
```

```
.....
```

```
</script>
```

```
...
```

```
<body>
```

```
....
```

```
=>
```

[TO-BE]

```
<script type="text/javascript" lazy="false">
```

```
    scwin.onpageload = function() {
```

```
    };
```

```
    scwin.onpageunload = function() {
```

```
    };
```

```
</script>
```

```
....
```

```
<body ev:onpageload="scwin.onpageload" ev:onpageunload="scwin.onpageunload">
```

```
.....
```

보다 효율적인 웹 브라우저의 메모리 관리를 위해서는 웹 페이지가 전환될 때 불필요한 자바스크립트 자원은 정리해야 합니다. 다시 말해, 일회성 자원과 재활용 자원 그리고 매번 로딩(실행)해야 하는 자원을 구분하고 일회성 자원의 경우는 페이지를 전환될 때 삭제해야 메모리가 누적되는 것을 방지할 수 있습니다. 아래와 같이 코딩 방식을 네임 스페이스 방식으로 작성하여 설정 기능을 이용합니다. config.xml의 spa설정의 variable속성에 정의한 변수명을 사용합니다.

기 존	SPA 적용 방식
<div>전역 영역에 할당하던 방식</div> <pre><script lazy="false"> var pageVal01 = "page01"; var pageVal02 = "page02"; var pageObj = { "title" : "Sample01", "author" : "User001" }; function getPage(){ return pageVal01; } function trigger1_onclick(){ return "onclick"; } </script></pre>	<div>네임 스페이스(name space) 방식</div> <pre><script lazy="false"> scwin.pageVal01 = "page01"; scwin.pageVal02 = "page02"; scwin.pageObj = { "title" : "Sample01", "author" : "User001" }; scwin.getPage = function(){ return scwin.pageVal01; }; scwin.trigger1_onclick = function(){ return "onclick"; }; scwin.onpageunload = function() { //페이지 전환 시 불필요한 자원 정리 }; </script></pre>

아래는 화면 개발을 하면서 생성되는 자바스크립트 객체들의 예시입니다.

주 최	종 류
웹스퀘어	<ul style="list-style-type: none">▪ UI 컴포넌트 객체▪ Data 객체▪ \$p를 통해 전역에 생성(할당)된 객체
사용자	<ul style="list-style-type: none">▪ 화면 내 script 블록 ex) <code>var pageVal01 = "page01";</code> <code>var pageObj = {</code> <code>"title" : "Sample01",</code> <code>"author" : "User001"</code> <code>};</code> <code>function getPage(){</code> <code>return pageVal01;</code> <code>}</code> <code>function trigger1_onclick(){</code> <code>return "onclick";</code> <code>}</code>▪ 화면 내 js 로딩 ex1) <code><script src="/ws5/js/common.js"/></code> ex2) <code>\$p.js("/ws5/js/common.js", "/ws5/js/module.js",function(){alert('done');});</code>▪ 공통 설정(config.xml) 내 js 로딩 ex) <code><engine><module src="/ws5/js/common.js" /></engine></code>

각각의 자원들을 아래와 같은 방법으로 정리할 수 있습니다.

위 치	종 류
화면 파일	<ul style="list-style-type: none">body의 onpageunload 이벤트 ex) <code><script type="text/javascript" lazy="false"><![CDATA[var pageTitle = "Sample01"; //onpageunload는 페이지가 전환되기 전에 실행 됩니다. scwin.onpageunload = function() { pageTitle = null; //자원 정리 };]]></script></code> <code><body ev:onpageunload="scwin.onpageunload"></code>
설정 파일	<ul style="list-style-type: none">config.xml의 spa 설정 <code><spa> <!-- spa로 페이지 전환 직전 호출할 함수의 변수명을 string으로 등록합니다 -> <onpageunload value=""/> <!-- spa로 페이지 전환 시 자동으로 삭제할 객체를 string형태로 등록합니다. --> <variable value="scwin,edu"/> <!-- spa로 count의 수만큼 페이지 이동 시 브라우저를 갱신합니다."/> --> <autoReload value="true" count="50"/> </spa></code>

페이지 전환 시 새로 로딩해야 하는 경우는 아래와 같이 설정할 수 있습니다.

위 치	종 류
화면 파일	<ul style="list-style-type: none">정적 로딩 ex) <script src="/ws5/js/common.js" cache="false"/>동적 로딩 ex) \$p.js("/ws5/js/common.js",function(){});
설정 파일	<ul style="list-style-type: none">config.xml의 설정 ex) <engine> <module src="/ws5/js/common.js" cache="false"/> </engine>

| 웹스퀘어 SPA

SPA(Single Page Application)는 브라우저에 웹 페이지를 좀 더 빠르게 표현하기 위한 방법론 중 하나입니다.
2003년도 이 개념이 논의되었으나 SPA의 어원은 명확하지 않습니다.

SPA는 웹 사이트를 사용할 때 화면의 전환없이 로딩된 자원을 재활용하여 변경된 부분만 적용하는 방식을 말합니다.
전통적인 동적 웹 페이지는 서버에서 결과물을 만들어내는 반면, SPA 방식은 브라우저에서 자바스크립트에 의해 결과물을 만듭니다.

적용되는 영역은 크게 다음과 같이 나눌 수 있습니다.

- 데이터 통신
- 페이지 변경

SPA를 적용할 때의 이점은 다음과 같습니다.

- 사용자의 사이트 이용의 흐름 유지
- 서버의 트래픽 감소
- 페이지간 빠른 전환

SPA를 적용할 때의 고려 사항은 다음과 같습니다.

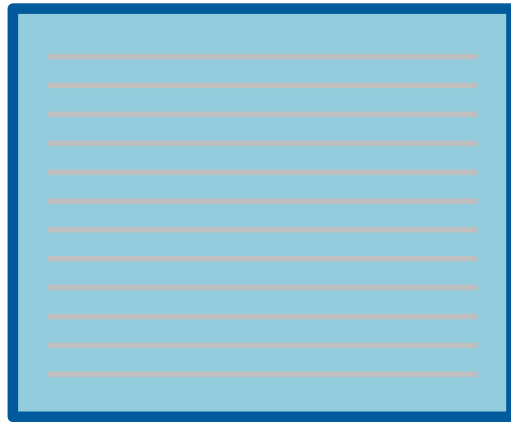
- 초기 자원 로딩 속도
- 페이지 전환 시 불필요한 자바스크립트 자원 해제

참조 : https://en.wikipedia.org/wiki/Single-page_application

웹스퀘어는 SPA를 크게 3단계에 걸쳐 적용하였습니다.
아래는 각 단계별 SPA 적용 범위에 대한 설명과 적용된 버전입니다.

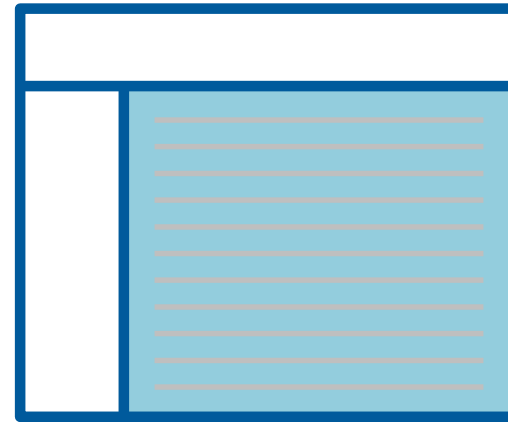
	설 명	적 용 버 전	비 고
1단계	데이터 통신 XHR(AJAX)	WebSquare WebSquare 2.0	▪ 2007년 출시
2단계	SPA IFrame에 SPA를 적용한 기능으로 IFrame을 재활용 합니다. 서버에서 화면 파일을 받아 스크립트로 변경되는 화면의 요소만 스크립트만 실행합니다. (SPA + IFrame)	WebSquare 5	▪ 2014년 출시
3단계	PAGE IFrame을 사용하지 않은 레이아웃 구성으로 WFrame을 사용하여 화면을 구성하고 여러 개의 콘텐츠가 추가되는 경우 Scope을 이용하여 객체명 중복을 방지합니다. (WFrame + Scope)	WebSquare 5 SP2	▪ 2017년 출시

- Form을 이용한 통신이 아닌 XHR(AJAX)로 서버와 데이터 통신을 합니다.



No Frame

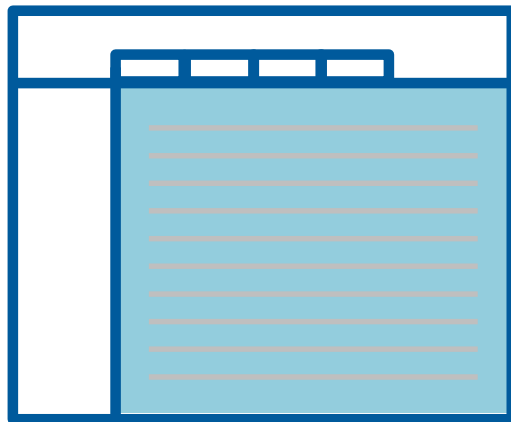
Single Document
Single View



Frame

Multiple Document
Single View

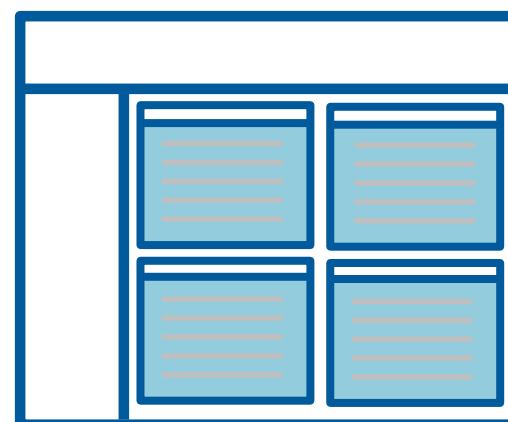
IFrame



Frame + Tab

Multiple Document
Single View

TabControl – IFrame mode



Frame + MDI

Multiple Document
Multiple View

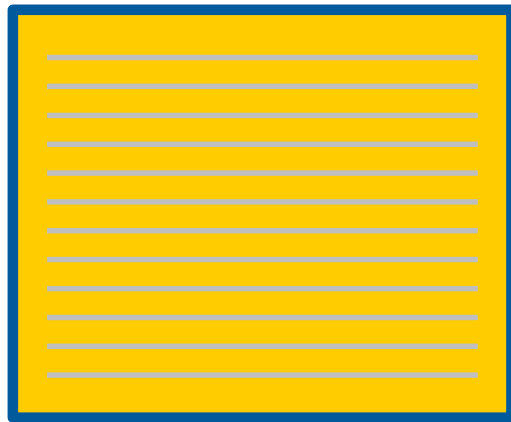
WindowConatiner - IFrame mode

다음은 SPA를 적용하기 위한 필수 설정입니다.

대 상	HTML	웹스퀘어
통신 API	<ul style="list-style-type: none">Form 사용 action에 정의한 URL를 통해 브라우저에 표현할 HTML을 서버에서 생성합니다.	<ul style="list-style-type: none">Submission 사용 XHR(AJAX)로 구성되어 있습니다. 데이터 String을 주고 받아 스크립트로 필요한 정보와 UI요소만 변경합니다.

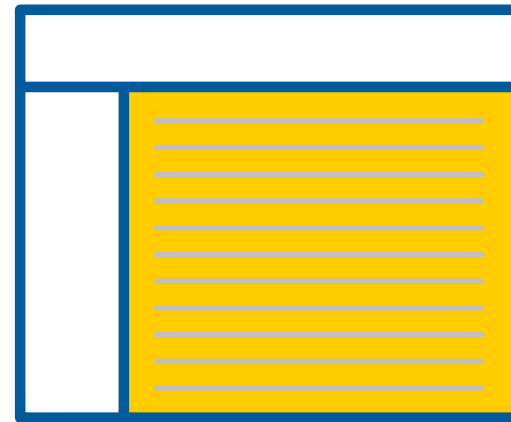
✓ 2단계 - SPA + IFrame

- 2014년도 출시된 웹스퀘어5 제품부터 지원되는 기능입니다.
- 브라우저의 새로 고침 없고 전환할 페이지 리소스만 받아 표현합니다. 화면 전환 시 제공된 API를 사용합니다.
- IFrame을 이용하는 동적으로 콘텐츠를 추가하는 컴포넌트의 경우, 초기에 IFrame을 미리 생성하여 재활용합니다.
- 동적으로 콘텐츠를 추가하는 경우 화면에 사용할 최대 IFrame의 개수를 한정 시켜야 합니다.



No Frame

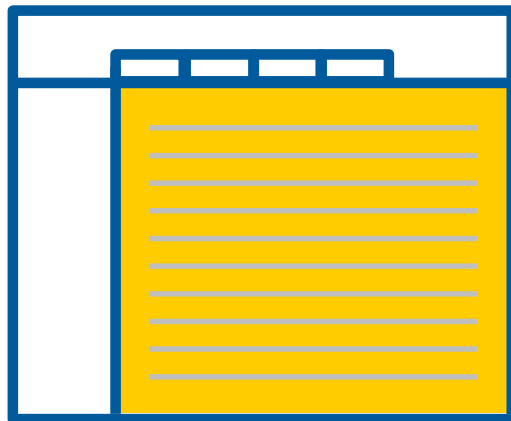
Single Document
Single View



Frame

Multiple Document
Single View

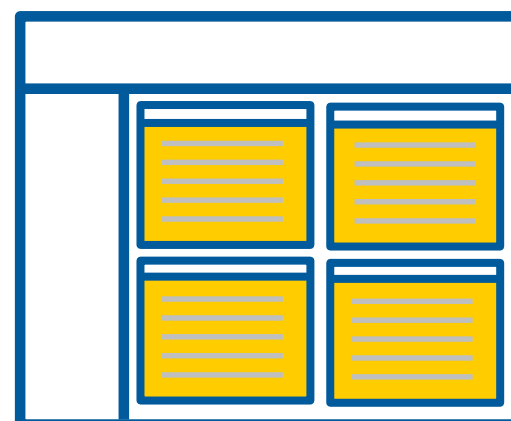
IFrame



Frame + Tab

Multiple Document
Single View

TabControl – IFrame mode



Frame + MDI

Multiple Document
Multiple View

WindowConatiner - IFrame mode

✓ 웹스퀘어 화면 파일 호출 방식

■ 기본 방식

쿼리 스트링(query string)을 통해 화면 파일의 경로를 받아 처리합니다.

이 때 사용하는 키(key)명은 "w2xPath" 입니다.

ex) `http://domain.com/websquare/websquare.html?w2xPath=/MA/MA01M01.xml`

■ SPA 방식(Hash 방식)

해시(hash) 방식으로 화면 파일의 경로를 전달합니다.

Hash 값은 기본 방식과 동일한 패턴으로 "w2xPath="뒤에 파일의 경로를 작성합니다.

ex) `http://domain.com/websquare/websquare.html#w2xPath=/MA/MA01M01.xml`

다음은 SPA를 적용하기 위한 필수 설정입니다.

대 상	기 존	SPA 적용 방식
화면 전환 API	<ul style="list-style-type: none">화면 전환 ex1) location.href = "....."; ex2) \$p.url("/ws5/page/A01.xml");	<ul style="list-style-type: none">화면 전환 \$p.url("/ws5/page/A01.xml", {spa:true});
Anchor	<ul style="list-style-type: none">화면 전환, 새 창으로 연결 ex1) <w2:anchor target="_self" href="/ws5/ws5.do?w2xPath=/ws5/page/A01.xml" /> ex2) <w2:anchor target="_blank" href="/ws5/ws5.do?w2xPath/ws5/page/A0001" />	<ul style="list-style-type: none">화면 전환, 새 창으로 연결 ex1) <w2:anchor target="_self" href="#w2xPath=/ws5/page/A01.xml"/> ex2) <w2:anchor target="_blank" href="/ws5/ws5.do#w2xPath=/ws5/page/A0001" />
IFrame	N/A	<ul style="list-style-type: none">spa 속성 spa로 화면을 전환 할 지 여부 ex) <w2:iframe spa="true" src="/ws5/page/A01.xml"/>
TabControl, WindowContainer	N/A	<ul style="list-style-type: none">spaInitCont 속성 몇 개의 탭(IFrame)을 미리 로딩할지 개수를 설정.windowMaxNum 속성 최대 생성할 탭(IFrame)의 개수 ex1) <w2:tabControl spaInitCount="10" windowMaxNum="10"/> ex2) <w2>windowContainer spaInitCount="10" windowMaxNum="10"/>

대 상	기 준	SPA 적용 방식
설정 파일 config.xml	N/A	<ul style="list-style-type: none">▪ spa/scriptCache 페이지 전환 시 JS를 다시 load하고 실행할지에 대한 여부.▪ spa/variable 페이지 전환 시 자동으로 삭제할 객체명. ex) <code><spa></code> <code><scriptCache value="true"/></code> <code><variable value="scwin"/></code> <code></spa></code>

분리된 파일을 화면에 포함할 때 사용하는 컴포넌트인 IFrame과 WFrame에 대한 간략한 설명입니다.
두개의 컴포넌트는 단일 컴포넌트로도 사용이 되지만 콘텐츠를 동적으로 추가하는 TabControl, WindowContainer, WidgetContainer와 같은 컴포넌트를 사용할 때도 파일을 어떤 방식으로 포함할지에 대한 옵션으로도 사용됩니다.

컴포넌트명	설 명
IFrame	<ul style="list-style-type: none">HTML의 IFrame과 동일한 기능을 제공합니다.웹스퀘어 페이지 이외 다른 페이지를 포함할 수 있습니다.IFrame로 구성된 페이지는 독자적인 영역이기 때문에 다른 페이지와의 간섭이 없습니다.
WFrame	<ul style="list-style-type: none">웹스퀘어 페이지만 포함할 수 있습니다.부모의 페이지에 병합(merge)되어 하나의 파일처럼 동작됩니다.<ul style="list-style-type: none">부모의 자원과 자식의 자원이 같은 영역에 위치됩니다.부모 페이지와 자식 페이지에 선언된 객체명이 중복되지 않아야 합니다.
WFrame + Scope	<ul style="list-style-type: none">링크된 페이지에 선언된 객체명들과 부모 페이지의 객체명이 중복될 경우 사용합니다.Wframe의 scope 속성을 true로 정의합니다.페이지별 scope을 적용하였으며 객체명 중복 방지를 위한 처리가 되어 있습니다.<ul style="list-style-type: none">런타임(실행) 시점에 웹스퀘어가 객체명을 재정의 합니다. (컴포넌트의 id를 변경합니다.)제공된 API를 통해 부모와 자식 간의 객체를 접근해야 합니다.페이지 개발 시 제시된 코딩법을 준수해야 합니다.

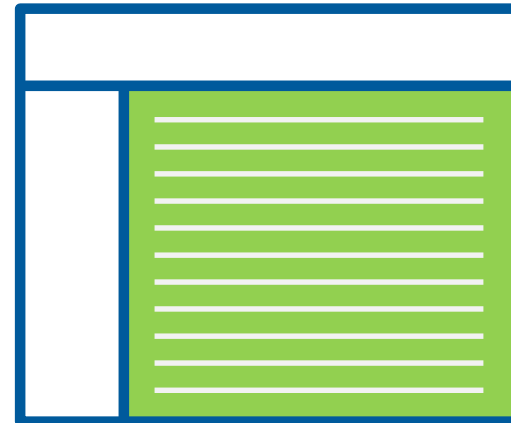
✓ 3단계 - SPA - WFrame + Scope

- 2017년도 상반기에 출시된 ServicePack2(SP2)부터 제공되는 기능입니다.
- 동적으로 콘텐츠를 추가하는 컴포넌트의 경우 IFrame이 아닌 WFrame(Div)을 이용하여 구성합니다.
- 전체 페이지가 하나의 websquare5 library를 사용합니다.
- WFrame을 사용하여 동적으로 추가된 page 간의 전역 객체명 중복 문제를 해결을 위해 Scope 옵션을 사용합니다.
- WFrame로 구성된 page의 경우 Scope이 적용되므로 websquare5가 제시하는 코딩법을 준수해야 합니다.



No Frame

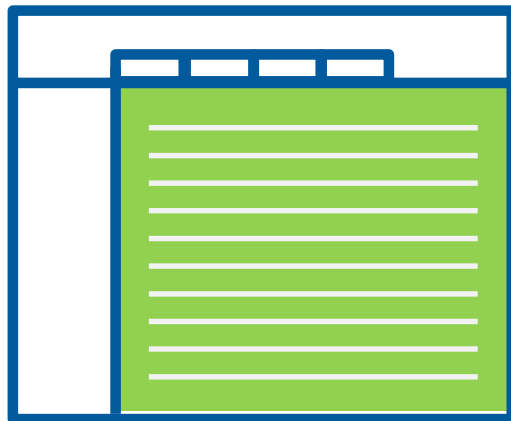
Single Document
Single View



Frame

Multiple Document
Single View

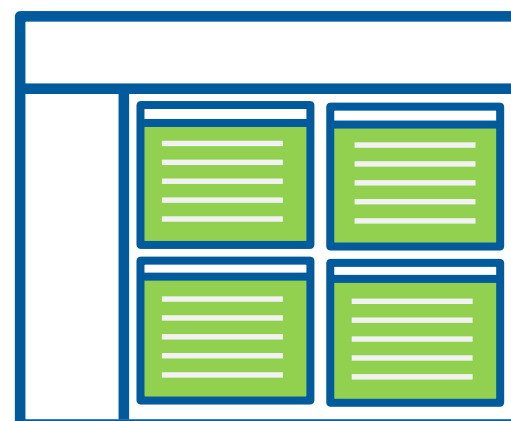
WFrame



Frame + Tab

Multiple Document
Single View

TabControl – WFrame mode



Frame + MDI

Multiple Document
Multiple View

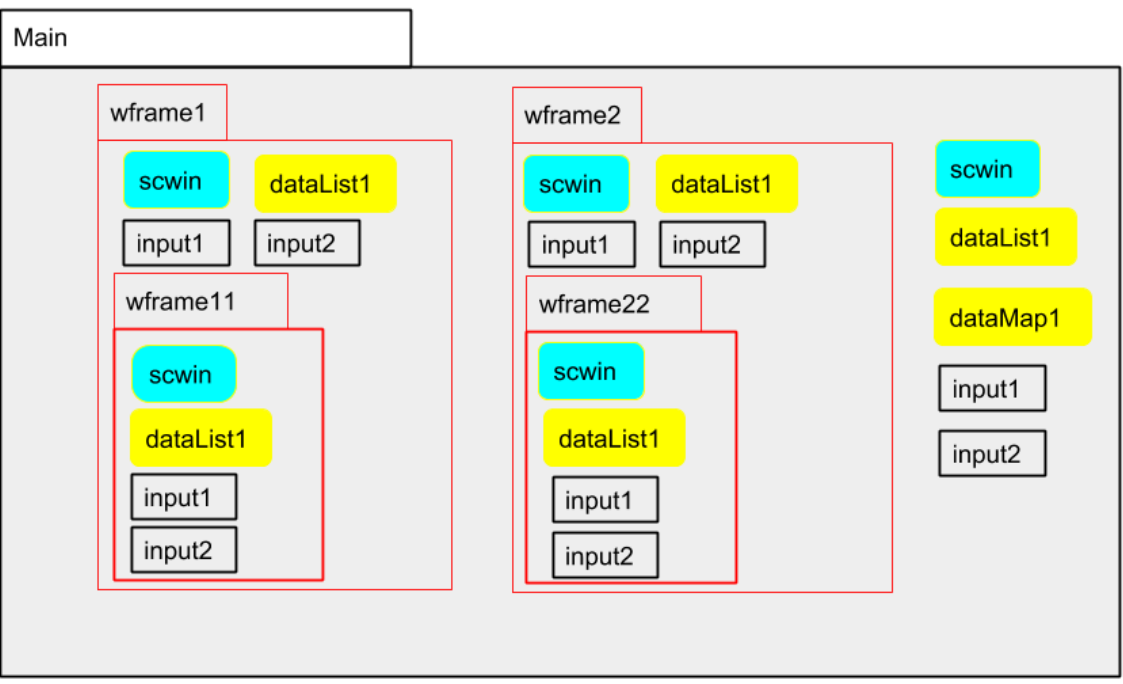
WindowConatiner - WFrame mode

다음은 SCOPE를 적용하기 위한 필수 설정 입니다.

대 상	SPA+IFrame 적용 방식	WFrame+Scope 적용 방식
화면 전환 API	<ul style="list-style-type: none">화면 전환 <code>\$p.url("/ws5/page/A01.xml",{spa:true});</code>	<ul style="list-style-type: none">동일
Anchor	<ul style="list-style-type: none">화면 전환, 새 창으로 연결 ex1) <code><w2:anchor target="_self" href="#w2xPath=/ws5/page/A01.xml"/></code> ex2) <code><w2:anchor target="_blank" href="/ws5/ws5.do#w2xPath=/ws5/page/A0001" /></code>	<ul style="list-style-type: none">동일
Frame	<ul style="list-style-type: none">IFrame	<ul style="list-style-type: none">WFrame의 Scope 속성 지정 ex1) <code><w2:wframe id="ui_contFrame" scope="true" ></w2:wframe></code>
TabControl	<ul style="list-style-type: none">spaInitCont 속성 몇 개의 탭(IFrame)을 미리 로딩할지 개수를 설정.windowMaxNum 속성 최대 생성할 탭(IFrame)의 개수 ex1) <code><w2:tabControl spaInitCount="10" windowMaxNum="10"/></code>	<ul style="list-style-type: none">addTab의 contentsOptions 속성 지정 ex1) <code>contentsOptions = { src : tabUrl, wframe : true, scope : true };</code> <code>[TabControl ID].addTab(tabKey, tabOptions, contentsOptions);</code>

대 상	SPA+IFrame 적용 방식	WFrame+Scope 적용 방식
WindowContaner	<ul style="list-style-type: none">▪ spaInitCont 속성 몇 개의 탭(IFrame)을 미리 로딩할지 개수를 설정.▪ windowMaxNum 속성 최대 생성할 탭(IFrame)의 개수 ex1) <w2:windowContainer spaInitCount="10" windowMaxNum="10"/>	<ul style="list-style-type: none">▪ frameMode 속성 ex1) <w2:windowContainer id="ui_windowContainer" frameMode="wframe"></w2:windowContainer>
설정 파일 config.xml	<ul style="list-style-type: none">▪ spa/scriptCache 페이지 전환 시 JS를 다시 load하고 실행할지에 대한 여부.▪ spa/variable 페이지 전환 시 자동으로 삭제할 객체명. ex) <spa> <scriptCache value="true"/> <variable value="scwin"/> </spa>	<ul style="list-style-type: none">▪ 동일

SCOPE Coding

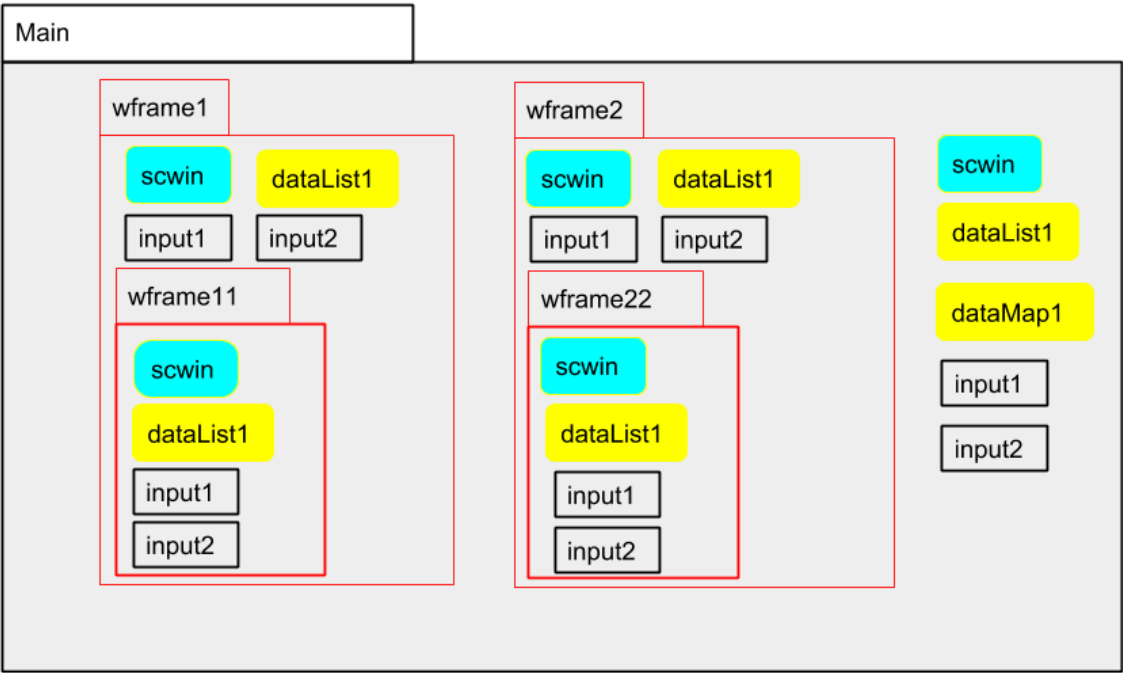


- 자식 페이지에서 부모 페이지로 접근할 때는 **\$p.parent()**를 사용합니다.
- 부모 페이지에서 자식 페이지를 접근할 때는 **wframe.getWindow();**로 scope객체를 얻어온 다음 scope객체를 통해 자식 페이지에 있는 객체를 접근합니다.

예시)

- 1) Main에서 wframe11의 input1 객체의 값을 가져오는 코드 : `wframe1.getWindow().wframe11.getWindow().input1.getValue();`
- 2) wframe22 에서 Main의 input2 객체에 값을 설정하는 코드 : `$p.parent().$p.parent().input2.setValue("test");`
- 3) wframe11에서 wframe22의 scwin.test(); 함수를 실행하는 방법 : `$p.parent().$p.parent().wframe2.getWindow().wframe22.getWindow().scwin.test();`
- 4) 자식 page에서 최상단의 부모 객체로 접근할 경우 `$p.top()` 을 통해서 접근 : `$p.top().input1.getValue();`

AliasDataCollection



- 자식 페이지에서 부모 페이지의 dataCollection 접근할 때 사용합니다.
- 동일한 type의 dataCollection에만 접근이 가능 합니다. (dataMap => dataMap, dataList =>dataList)
- 부모 페이지에서 자식 페이지의 dataCollection으로는 접근할 수 없습니다.

예시)

- 1) wframe11에서 wframe1 화면에 있는 dataList1을 참조하는 aliasDataCollection 정의
`<w2:aliasDataList id="aliasDataList1" scope="../dataList1"/> <!-- 상대경로 -->`
`<w2:aliasDataList id="aliasDataList1" scope="/wframe1/dataList1"/> <!-- 절대경로 -->`
- 2) wframe22에서 Main화면에 있는 dataMap1을 참조하는 aliasDataCollection 정의
`<w2:aliasDataMap id="aliasDataMap1" scope="../dataMap1"/> <!-- 상대경로 -->`
`<w2:aliasDataMap id="aliasDataMap1" scope="/dataMap1"/> <!-- 절대경로 -->`

✓ WFrame을 이용한 전체 레이아웃 구성 시 고려 사항

전체 레이아웃을 WFrame으로 구성한다는 의미는 1장의 웹스퀘어 페이지로 구성한다는 의미와 같습니다.
즉, 독립된 영역이 없이 같은 영역을 사용하기 때문에 아래와 같은 사항에 대해 고려해야 합니다.

- 화면갱신 및 Submission 실행 시 Layer 적용 범위
 - WFrame의 scope 부분에 대해서만 refresh 처리가 불가능합니다.
- CSS 작성시 고려사항
 - id selector 를 이용한 처리 불가 (component의 id가 변경되므로)
 - Selector에 따른 하위 영향도
 - 페이지별 CSS파일 분리 불가 (하나의 페이지로 인식되므로)
- 외부 모듈 연계 방법
 - 외부 모듈을 동일 페이지 내에 적용할 경우 해당 외부 모듈에 문제가 생기면 전체 페이지에 영향을 줍니다.
(IFRAME을 권장)

| Page Scope Coding

Page Scope를 이용한 Coding

child Page 확인

이름	<input type="text" value="순영후"/>	<input type="button" value="반영"/>	성별	<input checked="" type="radio"/> 남 <input type="radio"/> 여	<input type="button" value="반영"/>
입사일	<input type="text" value="2015-11-06"/>	<input type="button" value="반영"/>	직급	<input type="text" value="대리"/> ▼	<input type="button" value="반영"/>

wframe_tab

이 화면은 코드를 가져오기 위한 submission을 하나로 등록한 예시이다.
(성별 코드, 직위 코드)

성별 : ▼ 직급 : ▼

1,995건

NO	상태	사번	이름	성별	입사일	직급	이메일	우편번호	주소1	주소2	생성일	수정일
1		10000001	순영후	남 ▼	2015-11-06	대리 ▼	gevo206@inswave02.com	601802	부산 동구 범일1동	54~61	2016-12-07	2018-07-30
2		10000002	돈민상	여 ▼	2006-08-14	대리 ▼	tuyu121@inswave01.com	233808	강원 정선군 정선읍	덕우리	2016-12-07	2016-12-07
3		10000003	김해은	남 ▼	2002-06-10	팀장 ▼	wire062@inswave02.com	135898	서울 강남구 압구정1동	369~385	2016-12-07	2016-12-07
4		10000004	조성희	남 ▼	2004-10-08	차장 ▼	woka436@inswave03.com	700251	대구 중구 서문로1가		2016-12-07	2016-12-07
5		10000005	지다연	남 ▼	2016-10-11	팀장 ▼	ziku748@inswave01.com	135780	서울 강남구 신사동	도화종합기술공사	2016-12-07	2016-12-07

이름	<input type="text" value="순영후"/>	성별	<input checked="" type="radio"/> 남 <input type="radio"/> 여
입사일	<input type="text" value="2015-11-06"/>	직급	<input type="text" value="대리"/> ▼

TabControl 의 wframe scope 사용

Page Scope를 이용한 Coding

child Page 확인

이름

순영후

반영

성별

☒ 남 ☐ 여

반영

입사일

2015-11-06

반영

직급

대리

▼

반영

wframe_tab

이 화면은 코드를 가져오기 위한 submission을 하나로 등록한 예시이다.
(성별 코드, 직위 코드)

성별 :

-전제-

 ▼

직급 :

-전제-

 ▼

조회

1,995건

NO	상태	사번	이름	성별	입사일	직급	이메일	우편번호	주소1	주소2	생성일	수정일
1		10000001	순영후	남 ▼	2015-11-06	대리 ▼	gevo206@inswave02.com	601802	부산 동구 범일1동	54-1	2016-12-07	2018-07-30
2		10000002	돈민상	여 ▼	2006-08-14	대리 ▼	tuyu121@inswave01.com	233808	강원 정선군 정선읍	덕유리	2016-12-07	2016-12-07
3		10000003	감해은	남 ▼	2002-06-10	팀장 ▼	wire062@inswave02.com	135898	서울 강남구 압구정1동	369-85	2016-12-07	2016-12-07
4		10000004	조성희	남 ▼	2004-10-08	차장 ▼	woka436@inswave03.com	700251	대구 중구 서문로1가		2016-12-07	2016-12-07
5		10000005	지다연	남 ▼	2016-10-11	팀장 ▼	ziku748@inswave01.com	135780	서울 강남구 신사동	도화종합빌딩공사	2016-12-07	2016-12-07

이름

순영후

반영

성별

☒ 남 ☐ 여

반영

입사일

2015-11-06

반영

직급

대리

▼

반영

- Main Page에서 Tab의 WFrame에 접근 : tabControl의 id 기준으로 **getWindow**라는 API를 이용하여 접근
// wframe_tab 이라는 id를 가진 tabControl의 첫번째 tab안에 있는 ui_form_name 객체의 값을 가져오기
Ex) **wframe_tab.getWindow(0).ui_form_name.getValue()**;

Page Scope를 이용한 Coding

child Page 확인

이름순영후반영

성별남반영

입사일2015-11-06반영

직급대리반영

wframe_tab

이 화면은 코드를 가져오기 위한 submission을 하나로 등록한 예시이다.
(성별 코드, 직위 코드)

성별:-전제-

직급:-전제-

조회

1,995건

NO	상태	사번	이름	성별	입사일	직급	이메일	우편번호	주소1	주소2	생성일	수정일
1		10000001	순영후	남	2015-11-06	대리	gevo206@inswave02.com	1802	부산 동구 범일1동	54-61	2016-12-07	2018-07-30
2		10000002	돈민상	여	2006-08-14	대리	tuyu121@inswave01.com	3808	강원 정선군 정선읍	덕우리	2016-12-07	2016-12-07
3		10000003	감해은	남	2002-06-10	팀장	wire062@inswave02.com	5898	서울 강남구 압구정1동	369-385	2016-12-07	2016-12-07
4		10000004	조성희	남	2004-10-08	차장	woka436@inswave03.com	0251	대구 중구 서문로1가		2016-12-07	2016-12-07
5		10000005	지다연	남	2016-10-11	팀장	ziku748@inswave01.com	5780	서울 강남구 신사동	도화종합기술공사	2016-12-07	2016-12-07

이름순영후

성별남

입사일2015-11-06

직급대리

- Tab의 WFrame 에서 Main Page에 접근 : 부모 객체에 접근 할 때에는 **\$p.parent()** 라는 API를 이용하여 접근
// wframe_tab 이라는 id를 가진 tabControl의 첫번째 tab에서 부모 페이지의 p_name이라는 객체의 값에 데이터 넣기
Ex) **\$p.parent().p_name.setValue(dc_userInfoList.getCellData(0 , "EMP_NM"));**

✓ Page Scope – AliasDataCollection 사용

- Main Page에서 생성한 dataCollection 객체를 Page Scope 이 적용된 Page에서 사용할 경우 AliasDataCollection을 사용합니다.

The diagram illustrates the process of using an AliasDataCollection in a Page Scope. It shows three main components: the Main Page Data Info, the Scope Page Data Info, and the Property window for the AliasDataList.

Main Page Data Info: Shows a DataCollection with the following items:

- DataList#dc_codeGender
- DataList#dc_codePosition
- DataMap#dc_reqCodeParent

Scope Page Data Info: Shows a DataCollection with the following items:

- DataList#dc_userInfoList
- DataMap#dc_searchParam
- DataMap#dc_reqCode
- DataList#dc_code02
- DataList#dc_code101
- aliasDataList#aliasDc_codeGender
- aliasDataList#aliasDc_codePosition

Property window (AliasDataList): Shows the configuration for the AliasDataList. The 'scope' property is set to '../dc_codeGender', indicating it refers to the parent's DataCollection.

키	값
w2:aliasDataList	
id	aliasDc_codeGender
scope	../dc_codeGender
studioSrc	/ws5/training/complete/training3_1.xml
userData1	
userData2	
userData3	

부모의 dataCollection을 지정할 때 ../ 을 사용하여 위치를 지정합니다.

| 속도 향상을 위한 API

사용자가 로딩된 화면을 사용함에 있어 처리(반응) 속도를 향상 시키기 위한 방법과 컴포넌트의 API를 소개합니다.

- 통신 동기/비동기
 - Submission을 이용한 처리 : submission의 ondone 이벤트에서 처리함
 - Workflow를 이용한 처리 : workflow 편집모드에서 처리함
- 데이터 사이즈 축소
 - Data 처리 방식 비교 (예제: fn_dataFormat.xml, fn_largeData.xml)

방식	특징	format
XML	Data의 사이즈가 비교적 크고, 처리속도가 느림 Data의 손실이 발생할 수 있음 일반적인 웹에서 권장하지 않음	<list id="dlt_ZipCodeStreet"> <map> <SEQ_NO>1 </SEQ_NO> </map> </list>
JSON	일반적으로 웹에서 권장하는 방식	[{ "SEQ_NO":1, ...}, { "SEQ_NO":2 ... }]
1차원 Array	Data의 Size를 축소시켜주는 효과가 있으며, 가장 속도가 빠른 방법이지만, 코딩 방식에 어려움이 있음	"1","11110","2005001","세종대로
권장 Array	1차원 Array의 빠른 처리 속도를 유지함 코딩 방식에 장점을 가진 구조 대용량 처리에 적합한 방식	{ "columnInfo":["SEQ_NO",.....], "data":[1, "11110",.....] }

- GridView + DataList
 - CUD API (예제 : fn_gridView_CUD_API.xml)

DataList API 비교	전체	Row	Cell
insert		dataList.insertRow(idx) dataList.insertJSON(idx, jsonObj) dataList.insertXML(idx, xmlObj) dataList.insertData(idx, array)	
delete		dataList.deleteRow(idx) dataList.deleteRows(arrIdx)	
remove	dataList.removeAll() dataList.setJSON([]) // 빈값입력	dataList.removeRow(idx) dataList.removeRows(arrIdx)	
get	dataList.getAllJSON() dataList.getAllData() dataList.getAllXML()	dataList.getRowJSON() dataList.getRowXML() dataList.getRowData()	dataList.getCellData(rowIndex,colIndex)
set	dataList.setJSON(allJsonData)	dataList.setRowJSON() dataList.setRowXML() dataList.setRowData()	dataList.setCellData(rowIndex,colIndex,value)

- **dataList.setBroadcast** 설정을 사용하여 화면 갱신 여부를 설정할 수 있음.
- 컬럼 병합 - (예제 : fn_gridView_dynamicMerge.xml)
 - GridView의 body column의 colMerge="true" 속성을 사용
 - GridView의 mergeCells 속성을 사용 (byCol, byrow, bycolrec, byrowrec, byrestriccol, byrestricrow, never)

- FastScroll 속성 특징 (예제 : fn_gridView_fastScroll.xml)
 - 세로 스크롤을 이용하여 데이터를 조회 할 경우 렌더링하여 표현할 수 있는 최대한의 부분을 표시 후 미처 렌더링하지 못한 부분에 대하여는 메시지 처리해 주는 기능
 - 그리드뷰의 세로 스크롤을 보다 빠르게 렌더링하기 위해 사용하였으나 최근 브라우저의 성능이 좋아져 ie11 이상에서는 굳이 사용을 권장하지 않음

- Formatter 적용 특징 (예제 : fn_gridView_formatter.xml)
 - customFormatter를 통하여 gridView의 cell의 format을 변경할 수 있음
 - API를 이용하는 방법, innerHTML을 이용하는 방법

	장점	주의사항
API	엑셀 다운로드시 원본데이터로 다운받는다.	innerHTML방식 보다는 속도가 느리다.
innerHTML	렌더링 속도가 매우 빠르다	엑셀 다운로드시 변경된 데이터 포맷으로 다운받는다.

- IFrame/WFrame
 - IFrame – 일반적인 웹 페이지의 iframe 형태이다. 별도의 페이지 영역으로 주로 외부 모듈을 연계할 때 사용합니다.
 - WFrame – jsp의 include와 같은 개념으로 별도의 페이지 영역으로 간주되는 것이 아닌 하나의 페이지로 인식됩니다.
 - WFrame+Scope – WFrame내의 객체들을 별도의 scope객체 내에 감싸주어 적용
마치 WFrame을 IFrame처럼 사용할 수 있지만, scope 객체 접근 시 고유의 웹스퀘어5에서 제시하는 코딩 방법을 준수하여야 합니다.

✓ 속도 향상을 위한 API

- TabControl
 - 콘텐츠 구성 방법
 - IFrame Mode
 - WFrame Mode
 - WFramepreload Mode : alwaysDraw Mode 가 false 인 경우, 화면은 렌더링 하지 않고, 화면 xml만 읽어들이며 웹스퀘어 객체를 생성하기 까지만 수행하는 Mode
 - alwaysDraw 속성
 - true : tabControl이 가지고 있는 모든 탭의 컴포넌트를 렌더링
 - false : tabControl에서 활성화된 탭의 컴포넌트만 우선 렌더링 후 다른 탭이 활성화 될 때에 해당 탭 내의 컴포넌트를 렌더링
➔ true로 설정하면 초기 로딩이 오래 걸리므로 false 설정을 더 권장함
- Popup
 - Window popup
 - 별도의 팝업창으로 열리는 팝업
 - Layer popup
 - 페이지 내에서 열리는 layer 팝업
 - Floting Layer
 - WebSquare의 floatingLayer 라는 컴포넌트를 이용하여 show & hide 처리를 통하여 팝업처럼 보이게 하는 기능

✓ 성능에 대한 가이드 문서 위치

<https://wtech.inswave.kr> 접속 후 **Learning => Documentation** 선택

- [사용자 가이드]-[개발가이드]-[8. 성능 개선 가이드]
성능 향상을 위한 개발 방법에 대한 문서
- [사용자 가이드]-[SP4]-[파트 VI. 설정 가이드]
웹스퀘어 설정의 상세 가이드로 개발 환경 권장 설정과 운영 환경 권장 설정
- [사용자 가이드]-[SP4]-[파트 II. 페이지]
웹스퀘어5 SPA의 기능 및 설정 상세 가이드 및 WFrame + Scope을 이용한 코딩 방법

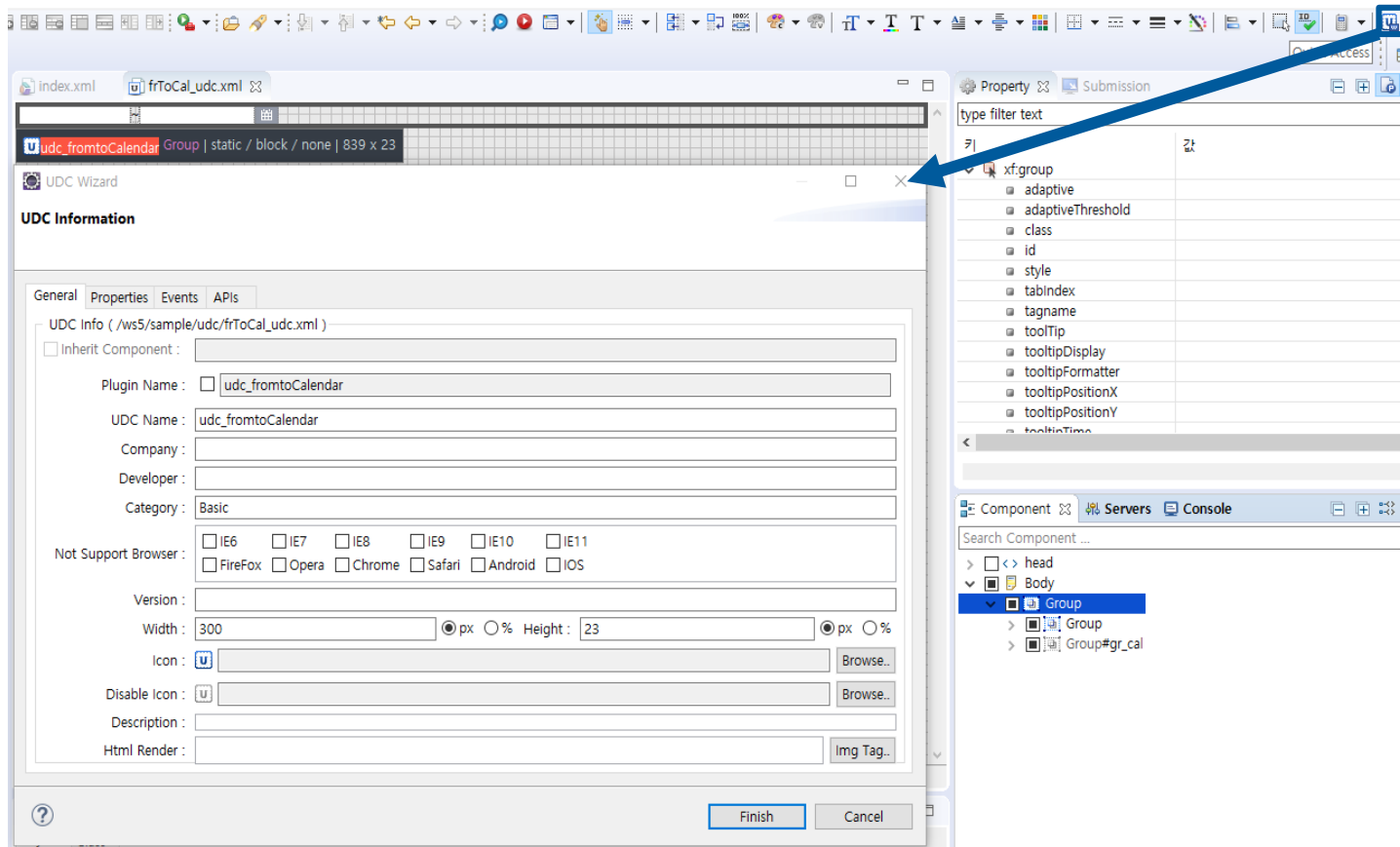
| 컴포넌트 활용

✓ 컴포넌트 활용

웹스퀘어는 다양한 기능들을 내포하고 있습니다.

그 중 문의가 많은 기능들에 대해 구현된 샘플을 통해 연동 방법을 안내하고자 합니다.

- UDC 컴포넌트 활용 (예제 : **frToCal_udc.xml**[udc파일], **fn_udcFromToCalendar.xml**[udc사용 파일])
 - 웹스퀘어5의 udc 객체를 생성하고 활용할 수 있습니다.
 - http://docs1.inswave.com/common_guides#20b8eb73abdb16e3 접속 후 13. UDC 가이드 참조



✓ 컴포넌트 활용

- 동적 컴포넌트 생성 (예제 : fn_dynamicCreate.xml)
 - \$p.dynamicCreate 라는 API를 이용하여 동적으로 컴포넌트를 생성할 수 있습니다.
- 엑셀 Provider (예제 : fn_largeData.xml)
 - AdvancedExceDownload의 옵션 중 provider 옵션을 이용하여 excel을 다운로드하는 방식
: provider 옵션으로 지정한 class에서 엑셀 조회 내용을 1차원 array로 return
 - AdvancedExceDownload의 옵션 중 Splitprovider 옵션을 이용하여 excel을 다운로드하는 방식
: splitProvider 옵션으로 내용을 분할하여 download 받을 수 있다. WAS의 out of Memory 발생을 줄여 줄 수 있습니다.
(websquare_home/config/websquare.xml에서 다운 받을 엑셀파일의 최대 row수 제한을 둘 수 있습니다.

maxRowCount 설정)

ex) <excel>

<download>

<!-- splitProvider 사용시 엑셀파일 최대row 수 설정, 미설정시 기본값 10만 -->

<maxRowCount value="500000" />

</download>

</excel>

✓ 컴포넌트 활용

- 다국어 처리
 - http://docs1.inswave.com/common_guides 접속 후 2. 다국어 설정가이드 참조
 - Client 다국어 처리 - fn_multiLang_client.xml
 - Server 다국어 처리 - fn_multiLang_server.xml
 - Properties 파일을 이용하는 방식
 - Database에 직접 table을 생성하여 data에서 처리하는 방식
- 엑셀 Image download (예제 : fn_gridView_excel_ImgDownload.xml)
 - gridView 에서 Image를 포함한 데이터가 있는 경우에 이를 엑셀로 내려주는 방법을 가이드 합니다.
 - advancedExcelDownload의 옵션 중 headerRowHeight, rowHeight, useImage 옵션을 필수로 주어야 합니다.
 - websquare_home -> config -> websquare.xml 에서 아래 옵션이 추가 되어야 합니다.
 - » <excel>
 - » <download>
 - » <baseDir value="서비스 기준경로(ROOT)" /> <!-- 예)
 - » <baseDir value="C:\WRM\WORKSPACE\ws5\WebContent\plugins\org.eclipse.wst.server.core\tmp1\wtpwebapps" /> -->
 - » </download>
 - » </excel>

※주의사항 :

- 1) 대용량에서는 적합하지 않으므로 사용하지 않습니다.
- 2) 이미지 경로가 was 서버에 존재해야 다운로드 할 수 있습니다.

이클립스의 로컬파일의 경로(C:\WRM\WORKSPACE\ws5\WebContent)가 아닌 wtp서버의 경로를 넣어야 합니다.

로컬파일의 경로를 넣을 경우, 웹스퀘어 화면에서 이미지 업로드후 다운로드 받을 때 새로 고침을 하지 않으면 실제로 업로드가 됐다고 인식하지 않아서 이미지를 내려 받을 수 없습니다.

감사합니다.