

 M2soft

 **CROWNIX Report 7**
more than the most

Crownix Report 7.0 Help

© 2020 Enter your company name

Table of Contents

소개	6
전제 조건	7
패키지 구성	8
시스템 요구사항	10
Quick Start	11
HTML5 Viewer UI	14
툴바	15
썸네일과 목차	16
페이지	17
스크롤	20
검색	22
보고서 실행방식에 따라 다른 검색 방식	23
검색 제약사항	24
사용자 정의 테마 구성	25
HTML5 Viewer 개발 가이드	26
보고서 실행하기	27
클라우드에서 실행하기	28
보안 모드	29
실행 모드	30
팝업(Popup)	31
임베드(Embed)	34
오버레이(Overlay)	37
Stateful & Stateless 모드	38
웹 폰트 사용하기	40
노트 사용하기	41
텍스트 노트 사용하기	43
인쇄하기	45
조회 툴바 사용하기	46
조회된 보고서와 데이터 임시보관	47
음성 지원 사용하기	49
PDF로 저장하기	51
서버 없이 폰트 포함 PDF 생성하기	52
TS A PDF 뷰어 사용하기	55
다이얼로그 사용하기	57
뷰어 문장 바꾸기	58
데이터 그리드 사용하기	60
보고서에 PDF를 삽입하여 사용하기	62
세로 스크롤 사용하기	63
.....	65
.....	66
.....	67
.....	71
.....	73
.....	76
.....	78

openFile(mrdPath, mrdParam, options)	79
openFile(mrdList, options)	80
options	81
노트 관련 속성	81
배율 관련 속성	84
UI 관련 속성	86
기타 속성	91
Page	96
nextPage()	97
prev Page()	98
movePage(pageNumber)	99
moveFirstPage()	100
moveLastPage()	101
moveFormField(FormFieldID)	102
hidePage(pageNumber)	103
showPage(pageNumber)	104
Zoom	105
zoomIn()	106
zoomOut([ratio])	107
pack()	109
Toolbar	110
ToolBarItem	111
hideToolBarItem(items)	112
showToolBarItem(items)	113
addToolBarItem(item)	114
hideNoteltem(items)	115
showNoteltem(items)	116
hideToolBar()	117
showToolBar()	118
dockToolBar(direction)	119
floatToolBar()	120
stickToolBar()	121
Export & Import	122
externalStorage(option, callback, openFileOpt)	123
export(type, callback)	127
exportFormData(options)	129
importFormData(formData)	132
MML Export & Import	134
exportMML(options)	135
importMML(mmlobject)	136
addSubmitParameters	138
bind(eventName, eventHandler)	140
cancel()	144
downloadFile(type)	145
getMrdPath()	147
getMrdParam()	148
getTotalPage()	149
print(options)	150
cloudPrint()	152
note()	153
cloudPrint()	155
readNfcTag()	156
scanBarcodes ()	158
setAlterFontMap	160

setDownloadFileName(fileName)	161
setStatelessMode(isStateless)	162
submit(url, datasource, servicecode, target, [csvfileName])	163
useAdjustLetterSpacing(used)	165
useAdjustCharacterScale(used)	166
useWebFont(used)	167
다이얼로그 관련 API	169
showFormDataStorageDialog()	170
showMMLStorageDialog()	171
showInquiryDialog()	172
showMovePageDialog()	173
Run-time 보고서 관련 API	174
addColumn(name, width, cellMergeOption)	175
addCell(index, title, variable, summarize, command, attribute, option, refField)	176
하이퍼링크 관련 API	179
nextDoc()	180
preDoc()	181
M2soft.crownix.Layout의 API	182
Theme	183
setTheme(theme)	184
setTheme(customTheme)	185
M2soft.crownix.Resource의 API	187
setLanguage(language)	188
M2soft.crownix.util.NumberUtil의 API	189
setLocaleFormat(locale, format)	190
M2soft.crownix.Util의 API	192
ActionEvents	193
m2soft.crownix.FormDialog의 API	195
show(contents)	196
HTML5 Viewer Callback Function 사용법	199
HLinkClicked(target, url, params)	200
보고서 작성 가이드	202
성능 관련 옵션	203
드릴다운 보고서 작성 팁	206
제약 사항	208
지원하지 않는 기능	208
뷰어 기능 제약	209
실행 환경에 의한 제약 사항	210
기타 환경에 의한 제약 사항	211
IE8 환경에서 지원하지 않는 API	216
트리그리드 보고서 제약 사항	217
틀고정 보고서 제약 사항	218
	220

Crownix HTML5 Viewer

<%PRODUCT%> 의 HTML5 Viewer (이하 HTML5 Viewer)를 실행하여 보고서를 조회하는 방법에 대하여 설명합니다.

소개

본 매뉴얼은 HTML5 Viewer의 설치 방법과 개발 방법을 설명합니다.

HTML5 Viewer는 ActiveX 등의 별도 설치 없이도 모든 웹 브라우저에서 동일하게 실행되는 크로스 브라우징 (Cross Browsing) 환경을 지원합니다.

또한 HTML5를 지원하는 다양한 데스크톱 및 모바일 디바이스에서 동작합니다.

전제 조건

HTML5 Viewer를 실행하기 위해서는 ERS Reporting Server가 설치되어 있어야 합니다.

HTML5 Viewer는 자바 스크립트 언어로 개발되었습니다.





본 매뉴얼에서 설명하는 샘플 코드는 자바 스크립트로 되어 있으므로 독자는 자바 스크립트 언어를 알고 있다고 가정합니다.

Reporting Server의 설치 방법은 ERS Reporting Server Installation 매뉴얼을 참조하시기 바랍니다.

패키지 구성

HTML5 Viewer의 패키지 및 파일 구조는 다음과 같습니다.

 js	- HTML5 Viewer를 동작 시키는데 필요한 자바 스크립트 파일들을 담고 있는 디렉토리
 formfield	- HTML5 Viewer의 폼필드를 별도로 사용하기 위한 라이브러리를 담고 있는 디렉토리
 crownix-formfield.min.js	
 invoke	- Reporting Server에 파일 저장 등의 요청을 하기 위한 라이브러리를 담고 있는 디렉토리
 crownix-invoker.min.js	
 share	- Form Sharing을 사용하기 위한 라이브러리를 담고 있는 디렉토리
 config.json	
 package.json	
 crownix-share-app.min.js	
 crownix-share-logger.min.js	
 crownix-share-server.min.js	
 crownix-viewer.min.js	- HTML5 Viewer 자바 스크립트(기본)
 crownix-viewer-ie8.min.js	- HTML5 Viewer 자바 스크립트(IE8 전용)
 crownix-viewer.storage.min.js	- HTML5 Viewer 자바 스크립트(보고서 저장 기능 포함)
 jquery-1.11.0.min.js	- HTML5 Viewer에서 사용하는 자바 스크립트 라이브러리
 css	- HTML5 Viewer를 동작 시키는데 필요한 CSS 파일들을 담고 있는 디렉토리
 crownix-viewer.min.css	- HTML5 Viewer의 CSS 파일
 sample	- 실행 예제 파일들을 담고 있는 디렉토리
 embed	- embed방식으로 실행되는 보고서 예제
 embed.html	
 ie8	- IE8 이하 브라우저를 지원하는 보고서 예제
 sample.html	
 invoker	- Reporting Server에 직접 요청하기 위한 invoker를 사용하는 예제
 invoker.html	
 invoker_download.html	
 invoker_downloadProtocolFile.html	
 invoker_print.html	
 invoker_print_downloadProtocolFile.html	
 overlay	- overlay방식으로 실행되는 보고서 예제
 overlay.html	

 popup	- popup방식으로 실행되는 보고서 예제
 popup.html	
 viewer.html	
 sample.html	- 기본 보고서 예제

HTML5 Viewer는 crownix-veiwewr-.min.js에 구현되어 있으며 jquery-1.11.0을 사용하여 개발되었습니다.

Viewer 실행 페이지에서는 jquery-1.11.0.min.js 와 crownix-veiwewr.min.js, crownix-viewer.min.css를 링크하여 사용합니다.

시스템 요구사항

- ERS Reporting Server

ERS Version	Java Engine
7.0.4.150 이상	7.0.4.73 이상

- 지원 브라우저

브라우저	지원 버전
 Internet Explorer	7.0 이상 (부분적으로 지원) 10.0 이상
 Chrome	4.0 이상
 Safari	4.0 이상
 FireFox	3.5 이상
 Opera	10.5 이상
 Android (Tablet, Mobile)	2.1 이상
 iOS Safari (Tablet, Mobile)	3.2 이상
상기 브라우저 외 HTML5의 Canvas 요소를 지원하는 모든 브라우저	

IE 10.0 미만 버전에서는 HTML5를 지원하지 않으므로 HTML5가 아닌 다른 방식으로 동작합니다. 따라서 HTML5로만 동작하는 일부 기능은 사용할 수 없습니다.

자세한 내용은 [IE8 환경에서 지원하지 않는 API](#) 와 [실행 환경에 의한 제약 사항](#)의 내용을 확인하십시오.

Quick Start

먼저 간단한 샘플을 통해 서버와 HTML5 Viewer가 원하는 위치에 제대로 설치되었는지와 참조하는 자바 스크립트 파일의 경로는 올바른지를 확인합니다.

아래의 몇 단계를 차례대로 거치면 마지막에 정상적인 화면이 나와야 합니다.

1 단계) ERS Reporting Server 설치

ERS Reporting Server 설치 방법은 ERS Reporting Server Installation Guide를 참고하십시오.

ü IE7/IE8 브라우저에서 보고서를 출력해야 할 경우

MobileServer/WEB-INF/lib 폴더에 있는 ers-mobile-handler.jar 파일을

ReportingServer/ WEB-INF/ lib 폴더로 복사합니다.

사용하는 라이선스는 Reporting Server 뿐 아니라 Mobile Server 도 지원해야 합니다.

2 단계) ERS Reporting Server의 charset 확인

<ERS Reporting Server 설치 디렉토리>/conf/server.properties의 charset의 값이 utf-8인지 확인합니다.

만약 utf-8이 아니면 utf-8로 변경해야 합니다.

```
...
server.request.charset = utf-8
server.response.charset = utf-8
...
```

3 단계) HTML5 Viewer의 설치

HTML5 Viewer 디렉토리 (제품에 포함된 html5viewer 폴더)를 WAS에 업로드 합니다.

톰캣을 WAS 로 사용한다면 Tomcat의 표준 Web Application 디렉토리인 TOMCAT_HOME/webapps에 파일을 복사합니다.(TOMCAT_HOME은Tomcat을 인스톨한 경로)

이와 같이 인스톨이 완료되면 WAS 를 재기동 합니다.

4 단계) HTML5 Viewer 패키지의 sample 실행

ReportingServer와 샘플 html 파일이 같은 장비에 있다면 sample/sample.html 파일을 브라우저로 엽니다.
ReportingServer가 다른 장비에 있다면 sample/sample.html 파일의 소스 코드를 아래와 같이 수정합니다.
http://localhost:8080/ReportingServer/service 코드를 해당하는 Server URL를 사용해서
<SERVER_URL>/ReportingServer/service 로 수정합니다.
파일을 저장하고 브라우저로 엽니다.

```
<!DOCTYPE html>
<html style="margin: 0; height: 100%">
<head>
  <meta charset="utf-8">
  <script src="../../js/jquery-1.11.0.min.js"></script>
  <script src="../../js/crownix-viewer.min.js"></script>
  <link rel="stylesheet" type="text/css" href="../../css/crownix-viewer.min.css">
</head>
<body style="margin: 0; height: 100%">
  <div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
  <script>
    window.onload = function(){
      var viewer = new
      m2soft.crownix.Viewer( 'http://localhost:8080/R
      eportingServer/service', 'crownix-viewer');
      viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    };
  </script>
</body>
</html>
```

IE8 브라우저에서 보고서를 출력해야 할 경우

sample/ie8/sample.html 파일에 있는 아래 코드를 참고합니다.

(브라우저가 IE8 이하일 경우 crownix-viewer-ie8.min.js 를 호출하도록 분기)

```
<!--[if !IE]><!-->
<script src="../../ js/ crownix-viewer.min.js"></ script>
<!--<![endif]-->
<!--[if gt IE 8]>
<script src="../../ js/ crownix-viewer.min.js"></ script>
<!--<![endif]-->
<!--[if lte IE 8]>
<script src="../../ js/ crownix-viewer-ie8.min.js"></ script>
<!--<![endif]-->
```

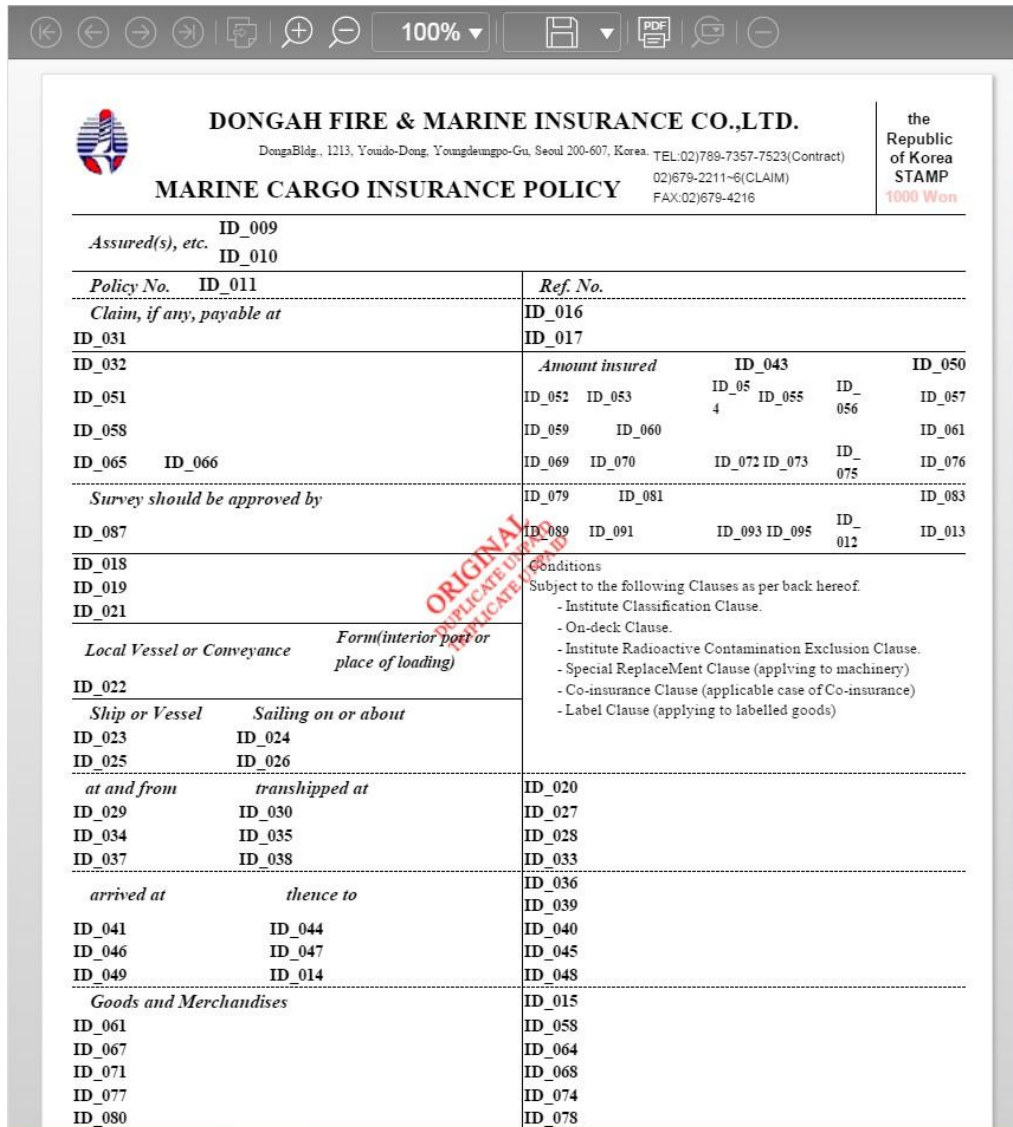
IE7 브라우저에서 보고서를 출력해야 할 경우

IE7 브라우저에서는 HTML5뷰어 툴바를 보기 위한 설정이 필요합니다.

```
<html xmlns:v="urn:schemas-microsoft-com:vml">
```

5 단계) sample 실행 결과 확인

sample.html을 실행하여 아래와 같은 보고서가 보이면 설치가 완료된 것 입니다.



<설치 확인용 sample.mrd를 HTML5 Viewer로 실행한 화면>

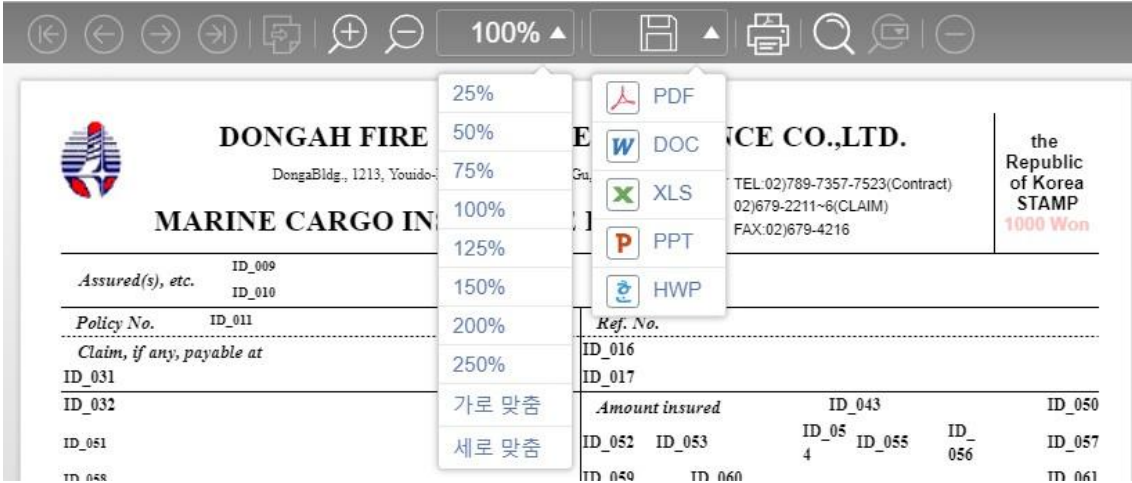
HTML5 Viewer UI

HTML5 Viewer의 UI를 설명합니다.

툴바

HTML5 Viewer를 실행하면 화면 상단에 다음 그림과 같이 기본 툴바가 표시됩니다.

API를 통해 툴바의 각 아이콘을 표시하거나 숨길 수 있습니다.



<기본 툴바>

툴바를 제어하기 위한 API 사용 방법 및 설명은 [HTML5 Viewer API 사용법](#)에서 [Toolbar](#)의 관련 내용을 참조하십시오.

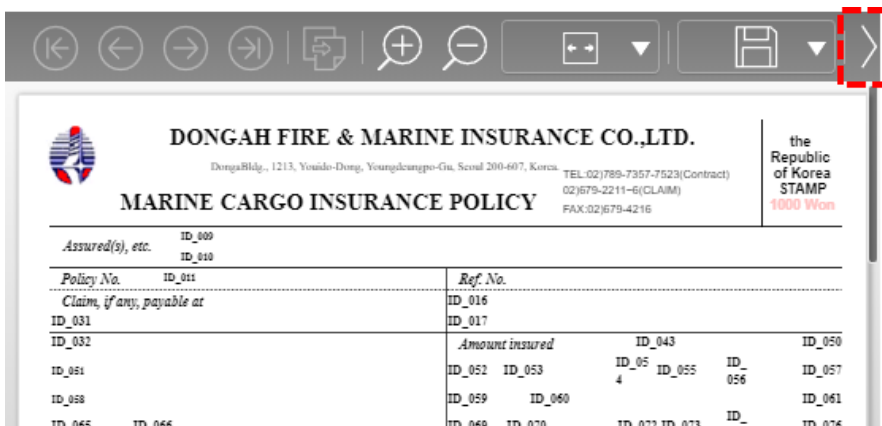
툴바를 구성하는 각 아이콘의 상세 기능은 [툴바 아이템](#)의 설명을 참조하십시오.

ü 툴바의 버튼이 화면보다 많아지는 경우?

이러한 경우 툴바의 좌우에 생기는 버튼이 나타납니다.

버튼을 누르면 메뉴가 스크롤 이동되어 메뉴를 선택할 수 있습니다.

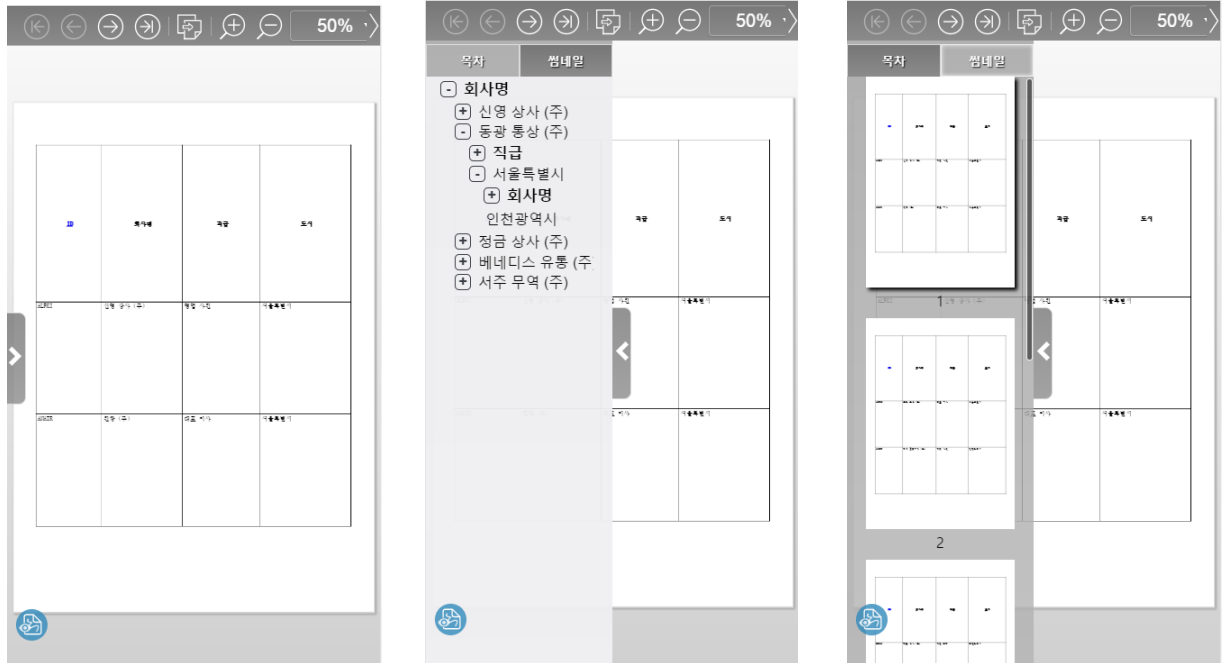
터치가 가능한 기기에서는 터치를 통하여 툴바를 좌우로 스크롤 하여 메뉴를 이동시킬 수 있습니다.



<툴바 스크롤 되는 상태>

썸네일과 목차

목차나 썸네일을 사용하는 환경에서는 다음과 같이 화면 왼쪽에 슬라이드 메뉴가 나타납니다.
이 메뉴에서 목차와 썸네일을 선택하여 표시할 수 있습니다.



- 목차를 통한 포커스 이동

목차의 항목을 선택하면 선택한 데이터가 있는 페이지의 해당 항목으로 이동합니다.

- 썸네일을 통한 페이지 이동

썸네일에서 페이지를 선택하면 선택한 페이지로 이동합니다.

- 제약사항

다이나믹 보고서를 사용하는 경우에는 썸네일과 목차에 적용되지 않습니다.

- 레이어를 사용하거나 폼필드 스크립트 또는 API를 사용하여 보고서 특정 영역이 변화한 경우
- 노트로 편집한 내용이 썸네일에 실시간으로 반영되지 않습니다.

목차 또는 썸네일을 사용하기 위한 API에 관한 자세한 설명은 [openFile API](#)의 [options](#)에서 [useTableOfContents](#) 와 [showThumbnail](#)의 설명을 참조하십시오.

페이지

페이지 영역에 보고서의 한 페이지를 보여줍니다.

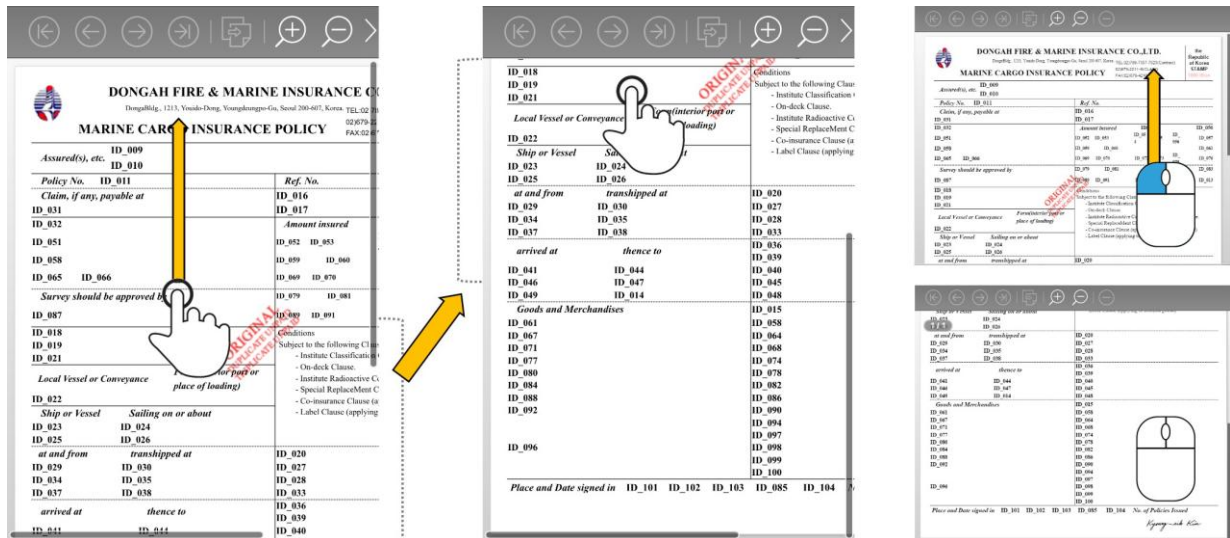
페이지를 보기 위해 제공하는 사용자 인터페이스는 다음과 같습니다.

- 드래그(Drag)
- 스크롤 휠 (Scroll Wheel)
- 핀치 줌 (Two Pinch Zoom)
- 더블 탭 (Double Tap)
- 페이지 넘김 (Page Swipe)
- 페이지 업 /다운 키(Page Up/Down Key)

• 드래그(Drag)

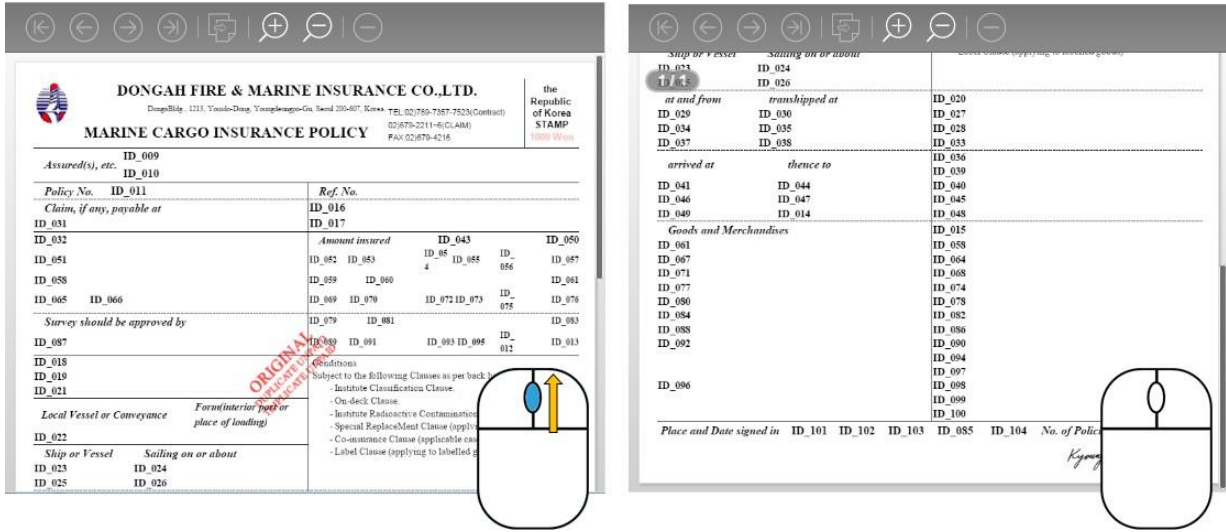
화면을 스크롤 할 때 사용합니다.

마우스로 페이지를 누르고 누른 상태로 움직이면 마우스를 따라 페이지가 이동합니다. 터치 스크린에서도 동일하게 동작합니다.



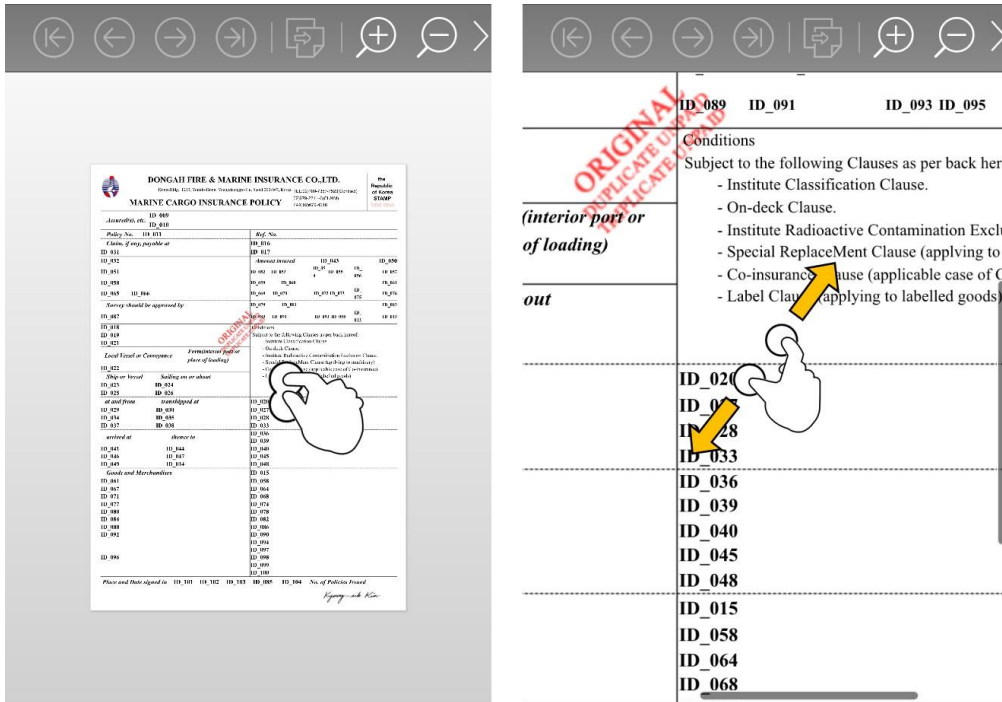
● 스크롤 휠(Scroll Wheel)

데스크 탑 환경에서 마우스 휠을 이용하여 문서를 위 아래로 스크롤 할 수 있습니다.



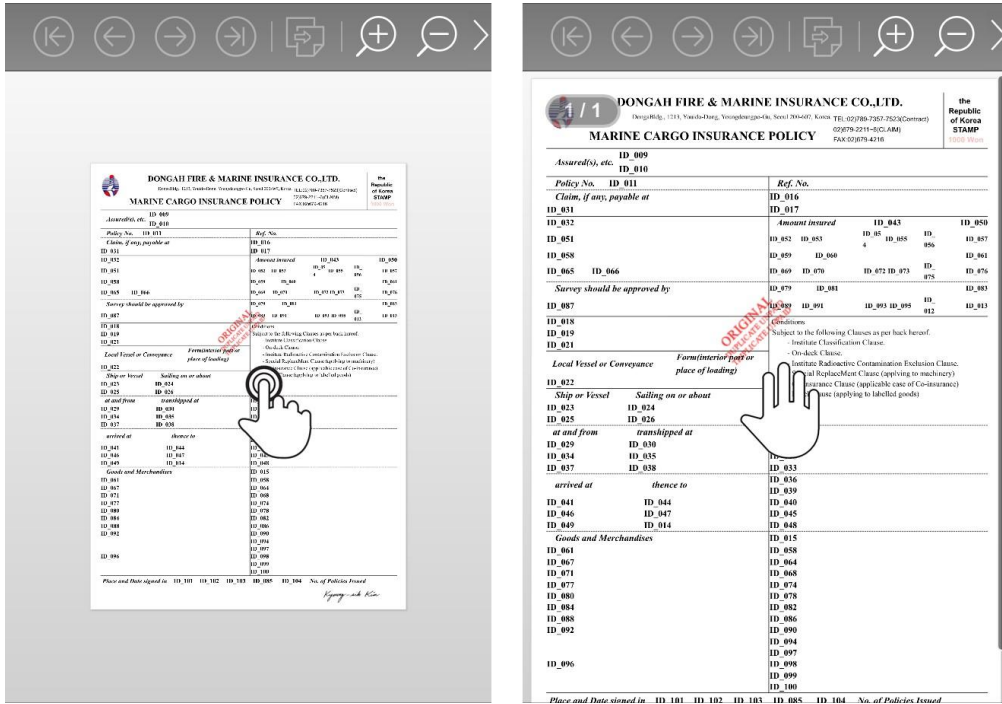
● 투 핀치 줌(Two Pinch Zoom)

터치 스크린에서 두 손가락을 화면에 대고 서로 멀어지거나 가까워지도록 하면 그에 맞게 문서가 확대되거나 축소됩니다.



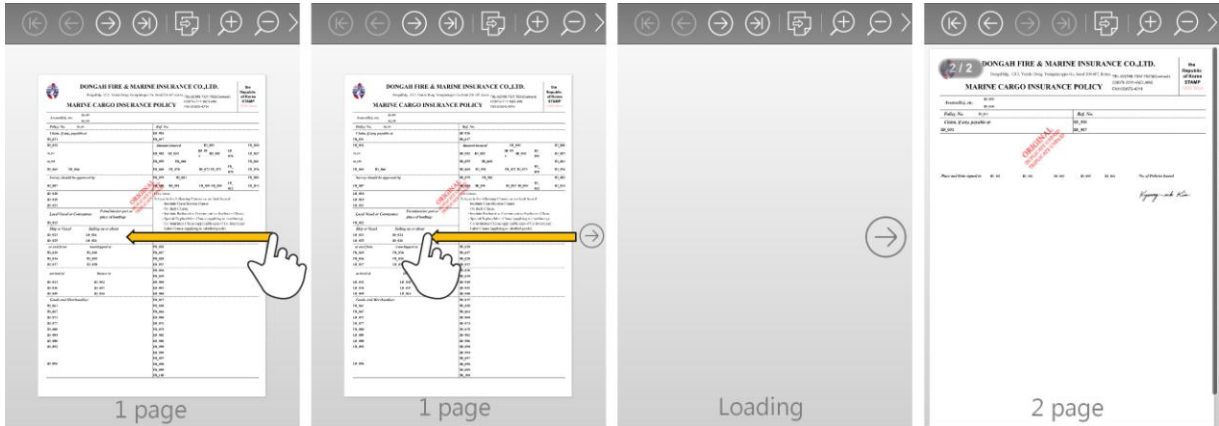
• 더블 탭(Double Tap)

터치 스크린 환경에서 페이지를 더블 탭 하면 화면이 확대되고 다시 더블 탭 하면 축소됩니다. 멀티 터치가 제공되지 않는 환경에서 유용합니다.



• 페이지 넘김(Page Swipe)

터치 스크린 환경에서 페이지를 좌우로 넘길 수 있습니다.



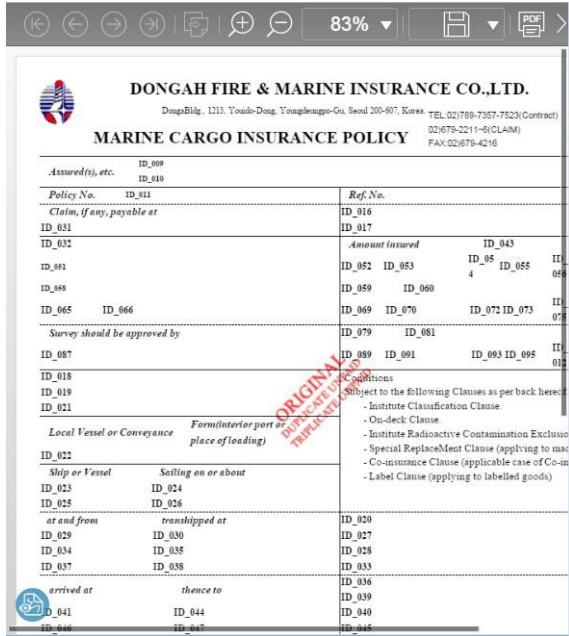
• 페이지 업/다운 키(PageUp/ Down Key)

데스크 탑 환경에서 페이지 업, 페이지 다운 키를 이용하여 문서를 위 아래로 스크롤 할 수 있습니다. 위 또는 아래로 스크롤 오버시 이전과 다음 페이지로 이동합니다.

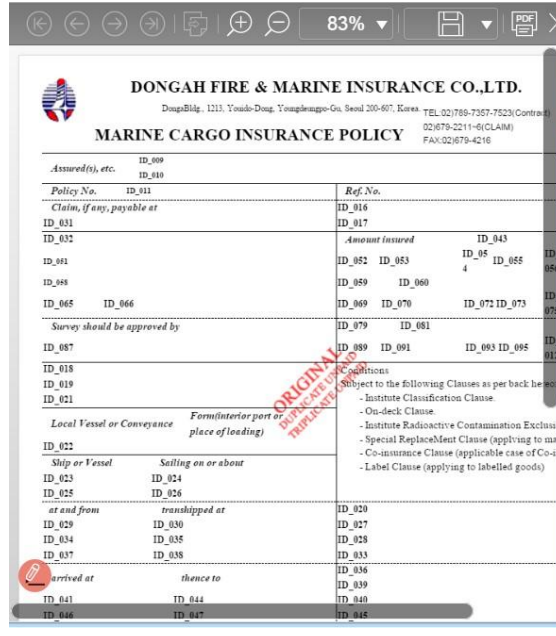
스크롤

HTML5 Viewer 화면 영역보다 보고서 페이지가 크면 상하 또는 좌우 스크롤바가 생깁니다.

일반적인 보고서 조회 모드일 때는 스크롤바 두께가 얇지만 노트 모드일 때는 스크롤바가 두꺼워집니다



<조회 모드 스크롤바>

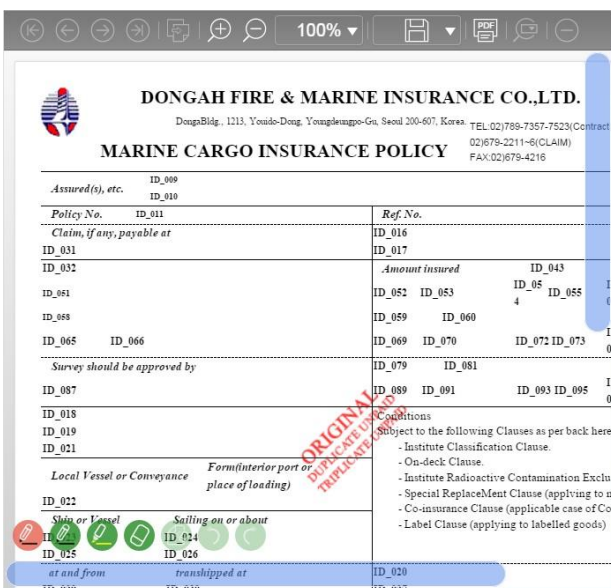


<노트 모드의 기본 스크롤바>

노트 모드에서 펜을 선택하여 그리는 동작과 일반적인 조회 모드에서 화면을 터치하여 스크롤하는 동작이 동일하기 때문에 노트 모드에서는 스크롤바를 사용합니다.

이러한 환경을 위하여 페이지 스크롤을 쉽게 조작할 수 있도록 스크롤바 색상이나 두께를 조정할 수 있습니다.

다음은 노트 모드의 스크롤바 두께와 색상을 지정한 경우의 스크롤 예시입니다.



노트 모드에서 `scrollbarOnNote` 프로퍼티를 사용하지 않으면 노트 모드의 기본 스크롤바 스타일로 표시됩니다. 스크롤바 스타일을 변경하고 싶으면 `scrollbarOnNote` 프로퍼티의 `width`와 `color`를 지정합니다.

노트 모드의 스크롤바 스타일 설정에 관한 자세한 설명은 `openFile`의 `option`의 속성 중 [scrollbarOnNote](#)의 설명을 참조하십시오.

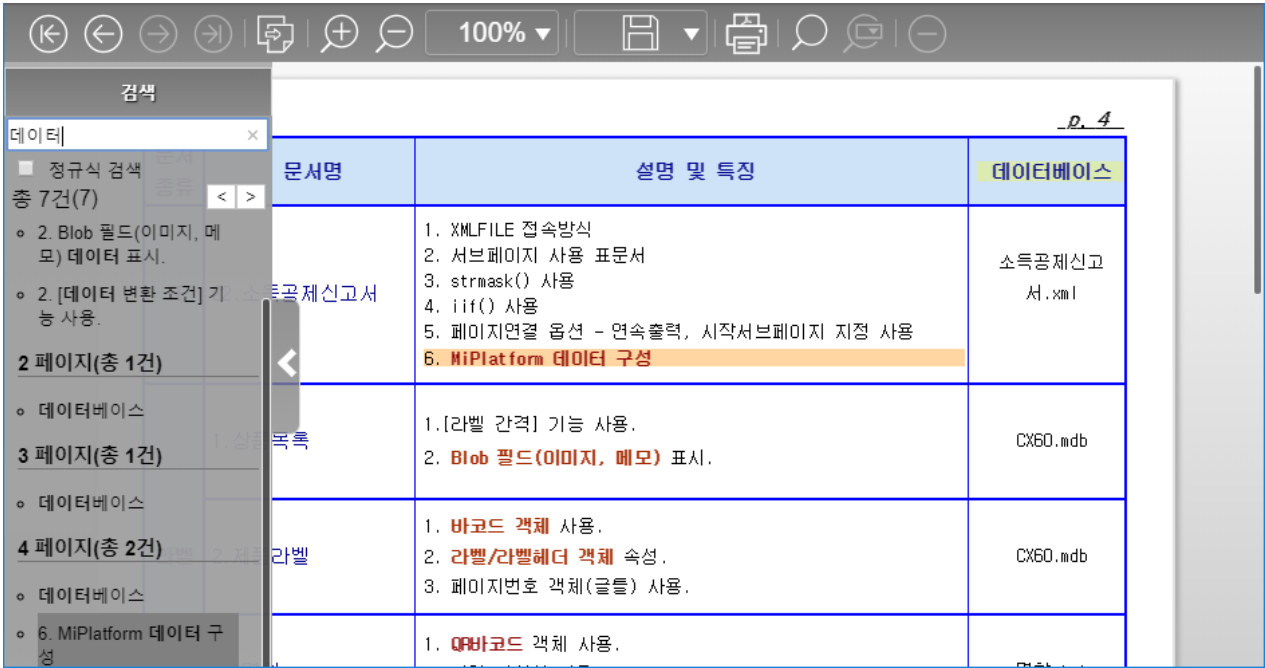
검색

HTML5 Viewer에서 찾고 싶은 내용을 검색할 수 있습니다.

검색한 내용은 페이지 단위로 정리되어 목차로 표시됩니다.

본문에서는 검색된 단어가 포함된 문장에 하이라이트를 적용합니다.

검색된 내용이 위치한 곳으로 이동하면 아래와 같이 구별된 하이라이트가 적용됩니다.



- 정규식 검색

정규식으로 검색할 때 사용합니다.

- 스마트 폼 보고서에서 검색되는 항목

호환 되지 않고 숨기기 상태가 아닌 텍스트 필드의 값

콤보박스에서 선택된 항목의 표시 데이터

파일이 첨부된 상태의 첨부 필드의 첨부 파일 명 버튼

명

체크박스과 라디오 버튼의 라벨

웹 필드의 가이드 문자열

보고서 실행방식에 따라 다른 검색 방식

HTML5 Viewer의 검색 방식은 서버사이드 방식과 클라이언트 방식이 있으며 각각의 검색 결과는 다를 수 있습니다.

뷰어에서 모든 페이지의 데이터를 받은 상태라면 클라이언트 방식으로 검색합니다.

뷰어에서 일부 페이지의 데이터만 받은 상태라면 모든 데이터를 보유한 서버에서 검색합니다.

모든 페이지 데이터를 받은 상태란?

Stateful 모드에서는 항상 모든 페이지 데이터를 받은 상태가 됩니다.

Stateless 모드에서는 기본적으로 일부 데이터(현재 조회중인 페이지의 데이터)만 받은 상태가 됩니다.

Statefull과 Stateless 모드에 대한 자세한 내용은 [Stateful & Stateless 모드](#)의 내용을 참조하십시오.

클라이언트 검색 방식

서버를 통하지 않고 뷰어에서 검색합니다.

검색어가 있는 페이지를 검색 목록에 표시하고 상세 검색 결과도 페이지 별로 표시합니다.

서버사이드 검색 방식에 비해 자세한 내용을 표시합니다.

서버사이드 검색 방식

검색 결과를 서버에서 받아서 표시합니다.

검색어가 있는 페이지를 검색 목록에 표시하고 상세 검색 결과는 검색 옵션에 따라 다르게 표시합니다.

서버사이드의 검색 옵션에 대한 자세한 설명은 [searchOption](#)의 내용을 참조하십시오.

검색 제약사항

텍스트 검색 제약

배분 정렬 스타일이 적용된 데이터는 한 글자 씩만 검색됩니다.

보고서 구조와 관련된 검색 제약

드롭다운이나 트리그리드와 같은 다이내믹 영역이 포함된 보고서에서 해당 영역의 데이터는 검색할 수 없습니다.

틀고정 보고서는 검색할 수 없습니다.

서버사이드 검색 방식의 제약

- 레이어가 포함된 보고서에서 레이어가 다른 보고서와 연결된 경우

클라이언트 검색방식에 한하여 검색이 가능하고 레이어가 포함된 페이지로 이동하여 실행한 다음부터 검색됩니다.

- 검색할 수 없는 항목

페이지 번호 객체

사용자 정의 테마 구성

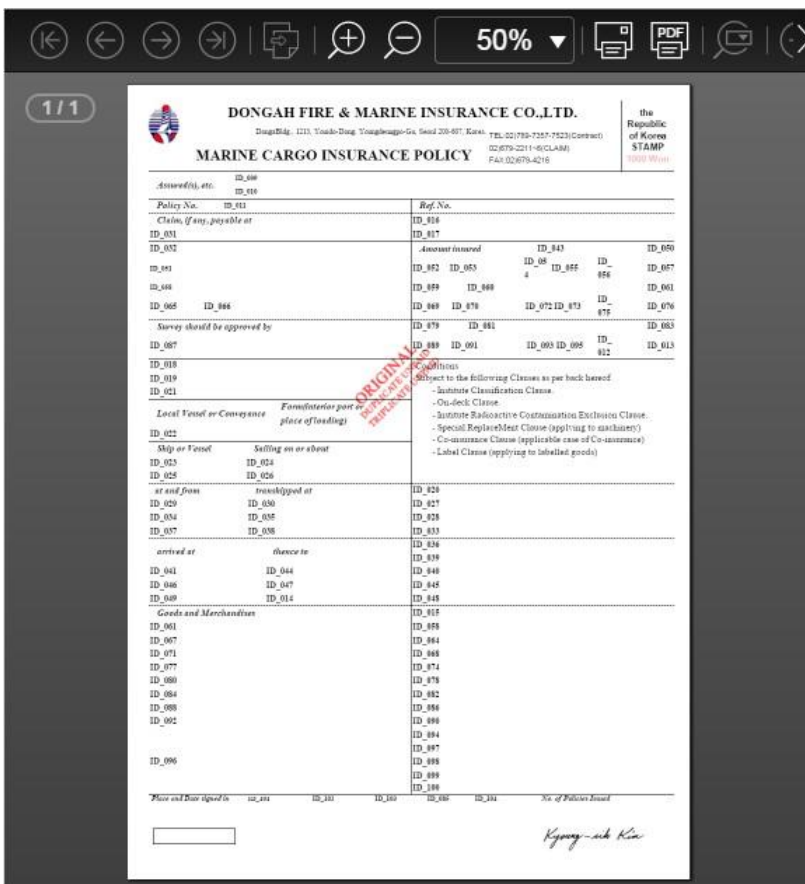
기본 툴바와 여백 페이지 및 색상을 변경하는 테마 기능을 제공합니다.

뷰어에서 기본적으로 제공하는 몇 가지 테마를 적용하거나 RGB 값을 직접 설정하여 자유롭게 색상을 변경할 수 있습니다

다음은 테마를 지정하는 스크립트 예시입니다.

```
<script>
window.onload = function(){
  m2soft.crownix.Layout.setTheme("Black");
  var viewer = new m2soft.crownix.Viewer(
    'http://localhost:8080/ReportingServer/service',
    'crownix-viewer');
  viewer.openFile('sample.mrd', '/rfn [sample.txt]');
};
</script>
```

위 스크립트를 실행하면 뷰어가 다음과 같은 색상으로 나타납니다.



테마의 설정에 관한 자세한 설명은 [setTheme](#)의 내용을 참조하십시오.

HTML5 Viewer 개발 가이드

HTML5 Viewer를 이용하여 보고서를 조회하기 위하여는 Java Script를 이용하여 뷰어를 구성해야 합니다. 이 장에서는 이와 관련된 개발 가이드를 설명합니다.

보고서 실행하기

[Quick Start](#) 세션에서 설명한 HTML 샘플을 변형하여 원하는 보고서를 실행할 수 있습니다.

[openFile](#) 메서드의 인자에 값을 설정합니다.

ü 첫 번째 인자 : MRD 파일의 이름 또는 경로를 설정

ü 두 번째 인자 : MRD 파일에서 사용하는 파라미터를 설정

```
<!DOCTYPE html>
<html style="margin: 0; height: 100%">
<head>
  <meta charset="utf-8">
  <script src="../../js/jquery-1.11.0.min.js"></script>
  <script src="../../js/crownix-viewer.min.js"></script>
  <link rel="stylesheet" type="text/css" href="../../css/crownix-viewer.min.css">
</head>
<body style="margin: 0; height: 100%">
  <div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
  <script>
    window.onload = function(){
      var viewer = new
        m2soft.crownix.Viewer( 'http://localhost:8080/ReportingServer/service', 'crownix-viewer');

      viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    };
  </script>
</body>
</html>
```

HTML 파일의 div 태그 영역의 설정을 통해 HTML5 Viewer의 화면이 표시됩니다.

위의 두 예제와 같이 width와 height를 100%로 설정하면 Viewer가 브라우저 화면을 꽉 채우게 됩니다.

width나 height 값을 조정하여 Viewer의 크기를 변경할 수 있습니다.

클라우드에서 실행하기

[보고서 실행하기](#)에서 ERS Reporting Server와 연결해서 보고서를 실행하는 방법을 소개했습니다. 이번에는 Crownix Cloud Reporting Service를 통해서 보고서를 실행하는 방법을 소개합니다.

- 방법

1단계) Crownix Cloud Reporting Service로부터 APIKey를 발급받습니다.

2단계) HTML5 Viewer로 보고서를 실행하기 전에 APIKey를 설정합니다.

3단계) HTML5 Viewer로 보고서를 실행합니다.

```
m2soft.crownix.Ajax.xApiKey = <Crownix Cloud Reporting Service 에서 발급된 API Key>; // 문자열  
var viewer = new m2soft.crownix.Viewer('https://cloud.crownix.com/service', 'crownix_viewer');  
viewer.openFile('sample.mrd', '/rv test [test data]');
```

보안 모드

HTML5 Viewer는 HTTPS 환경을 권장하므로 기본적으로 데이터 보안 모드를 제공하지 않습니다. 그러나 HTTPS 환경을 사용하지 않는 사용자의 경우 서버와 통신하는 데이터를 암호화하여 실행할 수 있습니다. 이 때 Reporting Server에 `secure.enable`이 `true`로 적용되어 있어야 하며 폼 필드 기능을 사용할 경우 InfoTalk Server에도 `secure.enable`을 `true`로 설정을 해야 합니다.

관련 설정은 [\[ERS Reporting Server Installation\]](#), [\[ERS InfoTalk Server Installation\]](#) 매뉴얼을 참조하시기 바랍니다.

HTML5 Viewer에서 보안 모드를 설정하기 위하여는 [openFile](#) 메서드의 인수 중 `options`의 `enableSecure` 키를 `true`로 설정하여 적용합니다.

```
<script>
window.onload = function(){
  var viewer = new
  m2soft.crownix.Viewer( 'http://localhost:8080/R
  eportingServer/service', 'crownix-viewer');

  viewer.openFile(
    'sample.mrd', '/rfn [sample.txt]',
    { enableSecure: true });
  };
</script>
```

실행 모드

HTML5 Viewer는 Popup/Overlay/Embed의 세 가지 실행 모드가 있습니다

팝업(Popup)

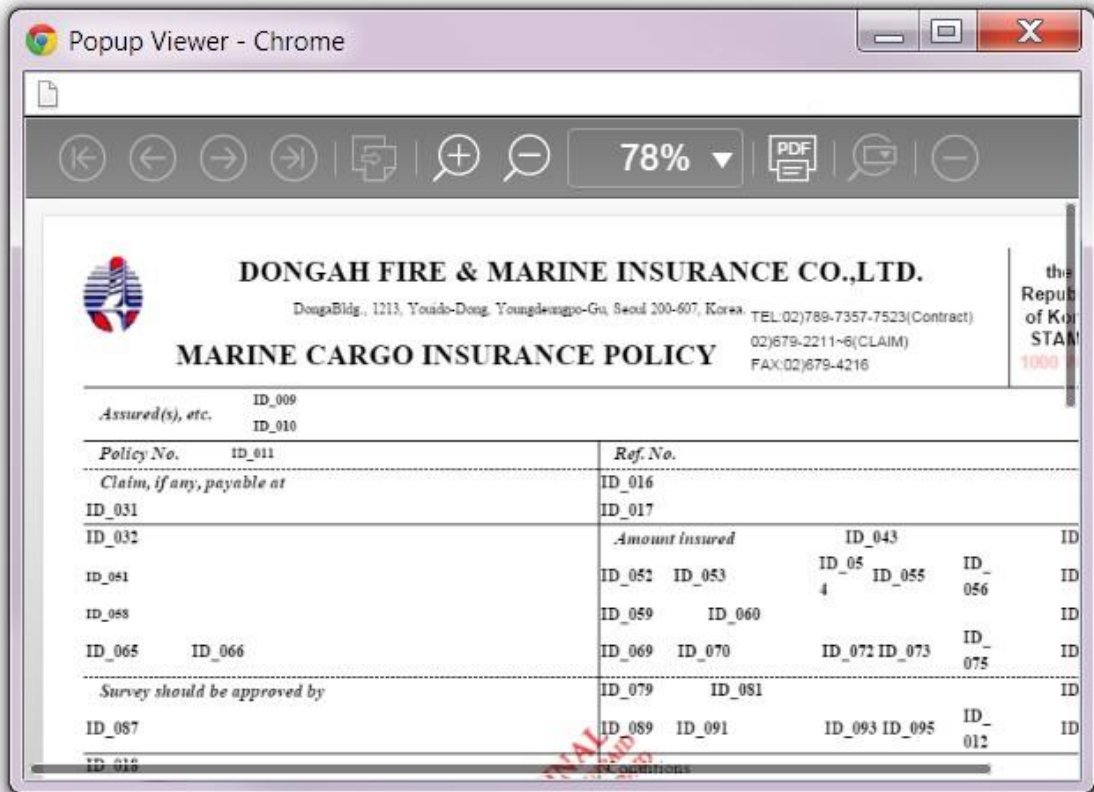
웹 페이지의 버튼을 클릭했을 때 HTML5 Viewer가 팝업 창으로 실행되는 방식입니다.

이 방식은 기존 웹 페이지의 변경이 적고 Viewer 화면을 크게 볼 수 있는 것이 장점입니다.

URL을 이용해 파일명과 파라미터를 넘기므로 URL의 길이 제한을 넘는 파라미터와 파일명은 사용할 수 없습니다.

URL의 길이 제한은 브라우저의 스펙을 따르며 각 브라우저마다 상이합니다.

execute



<팝업 방식으로 실행한 화면>

- 팝업 방식으로 실행하기 위한 샘플 코드

HTML5 Viewer 패키지의 샘플 중 popup.html 샘플을 참조하십시오.

```

<!DOCTYPE html>
<html style="margin: 0; height: 100%">
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0,
    maximum-scale=1.0, minimum-scale=1.0, user-scalable=no " />
  <meta charset="utf-8">
  <script src="../js/jquery-1.11.0.min.js"></script>
  <script type="text/javascript">
    $(document).ready(function(){
      $('#popup').click(function(){
        // 웹 페이지의 버튼을 클릭하면 window.open() 함수를 호출합니다.
        window.open(
          './viewer.html?mrd_path=sample.mrd&mrd_param=/rfn [sample.txt]
          & server_url=http://localhost:8080/ReportingServer/service',
          'ers_viewer', 'width=900,height=800');
      });
    });
  </script>
</head>
<body style="margin: 0; height: 100%">
  <input type="button" id="popup" value="execute" />
</body>
</html>

```

- window.open() 설정 방법

첫 번째 인자

Viewer와 실행할 보고서 명과 파라미터와 서버 주소를 URL Query 형식에 맞게 작성해야 합니다.

```
./viewer.html?mrd_path=<mrd path>& mrd_param=<mrd param>server_url=<server
address>/service
```

두 번째 인자

팝업 창의 이름을 입력합니다.

window.open이 호출될 때 같은 이름의 창이 떠있으면 새로 팝업 창이 뜨지 않고 열려 있는 창을 사용 합니다.

그러므로 위의 샘플 페이지의 execute 버튼을 여러 번 누르더라도 여러 개의 창이 뜨지 않고 하나의 팝업 창을 재사용하게 됩니다.

- 팝업 창에 표시 되는 viewer.html 샘플

샘플 디렉토리에서 참조하십시오.

페이지가 로드되면 이전 웹 페이지로부터 넘겨 받은 mrd_path, mrd_param, server_url 정보를 이용해 Viewer를 실행합니다.

이 예제에서 Viewer 영역은 팝업 창의 전체 크기입니다.

HTML5/CSS 코딩을 이용해 팝업 창 화면을 원하는 형식으로 꾸밀 수 있습니다.

```
<!DOCTYPE html>
<html style="margin: 0; height: 100%">
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0,
    maximum-scale=1.0, minimum-scale=1.0, user-scalable=no " />
  <meta name="apple-mobile-web-app-capable" content="yes">
  <meta name="apple-touch-fullscreen" content="yes">
  <meta charset="utf-8">
  <script src="../../js/jquery-1.11.0.min.js"></script>
  <script src="../../js/crownix-viewer.min.js"></script>
  <link rel="stylesheet" type="text/css" href="../../css/crownix-viewer.min.css">
</head>
<body style="margin: 0; height: 100%">
  <div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
  <script>
    $(document).ready(function(){
      var serverUrl = $.getUrlVar('server_url');
      var mrdPath = $.getUrlVar('mrd_path');
      var mrdParam = $.getUrlVar('mrd_param');

      var viewer = new m2soft.crownix.Viewer(serverUrl, 'crownix-viewer');
      viewer.openFile(mrdPath, mrdParam);
    });
  </script>
</body>
</html>
```

임베드(Embed)

웹 페이지의 일부 영역에 HTML5 Viewer가 포함되는 방식입니다.

Viewer의 위치와 크기를 지정할 수 있습니다.

아래는 임베드 방식으로 실행되는 HTML5 Viewer의 예입니다.

The screenshot shows a web page for M2 Soft Co., Ltd. with a navigation menu (Company, Solution, Press Center, Clients, Support, Demonstration) and language options (JAPANESE, KOREA). Below the menu is a banner for 'Customer First Philosophy built on the Most Optimized Solution Business Intelligence/Reporting Solution'. The main content area features a 'DEMO' section with a 'GO Home' button and an embedded HTML5 Viewer. The viewer displays a document titled 'MARINE CARGO INSURANCE POLICY' from Dongah Fire & Marine Insurance Co., Ltd. The document includes a table with various ID fields and a 'Stamp' from the Republic of Korea.

Assured(s), etc.		Policy No.		Ref. No.	
ID_009	ID_010	ID_011	ID_016	ID_017	
Claim, if any, payable at		Amount insured			
ID_031	ID_032	ID_051	ID_052	ID_053	ID_054
ID_058	ID_065	ID_066	ID_059	ID_060	ID_061
ID_065	ID_066	ID_069	ID_070	ID_072	ID_073
Survey should be approved by		ID_079	ID_081	ID_075	ID_076
ID_087	ID_018	ID_089	ID_091	ID_093	ID_095
ID_018		ID_093	ID_095	ID_012	ID_013

<임베드 방식의 예시>

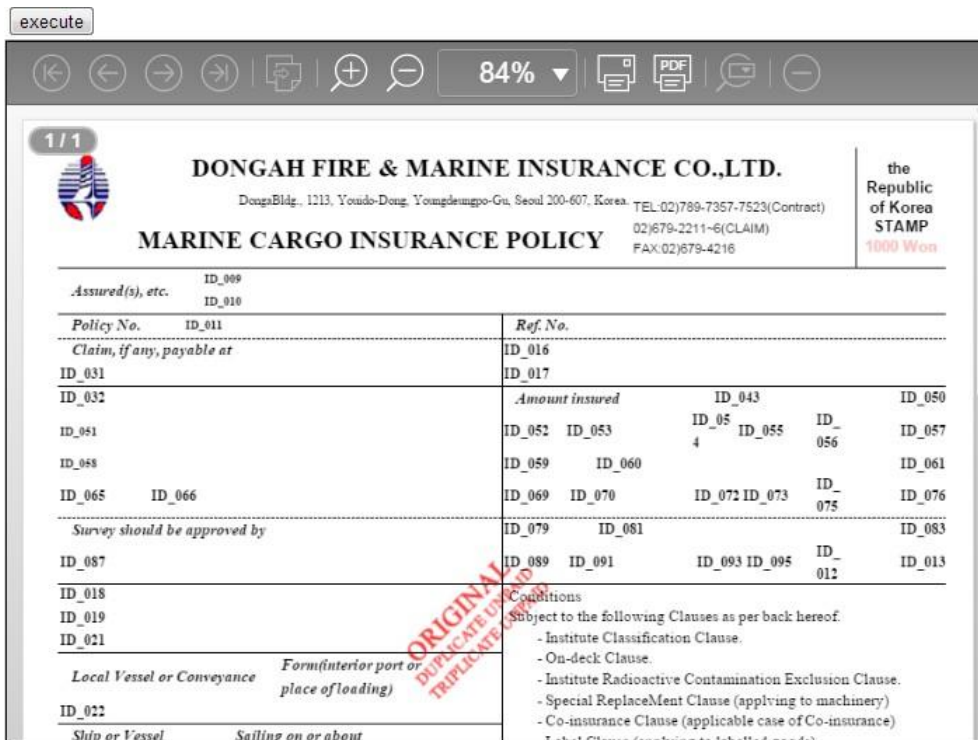
임베드 방식으로 실행하기 위한 샘플 코드는 아래와 같습니다.

임베드 방식은 Viewer의 레이아웃이 전체 화면이 아닌 것 외에는 팝업 방식과 사용이 비슷합니다.

HTML5 Viewer 패키지의 샘플 중 `embed/embed.html` 샘플을 참조하십시오.

```
<!DOCTYPE html>
<html style="margin: 0; height: 100%">
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0,
    maximum-scale=1.0, minimum-scale=1.0, user-scalable=no " />
  <meta charset="utf-8">
  <script src="../../js/jquery-1.11.0.min.js"></script>
  <script src="../../js/crownix-viewer.min.js"></script>
  <link rel="stylesheet" type="text/css" href="../../css/crownix-viewer.min.css">
  <script type="text/javascript">
    $(document).ready(function(){
      var viewer = new m2soft.crownix.Viewer(serverUrl, 'crownix-viewer');
      $('#embed').click(function(){
        viewer.open('sample.mrd', '/rfn [sample.txt]');
      });
    });
  </script>
</head>
<body style="margin: 0; height: 100%">
  <input type="button" id="embed" value="execute" />
  <div id="crownix-viewer" style="position:absolute; width:700px; height:500px; border:1px solid #000;"></div>
</body>
</html>
```

아래 그림은 위 샘플 코드에 대한 실행 화면입니다.



<임베드 방식으로 실행한 샘플 화면>

div 태그 내의 style=position 속성을 absolute나 relative로 설정하면 임베드 방식으로 실행됩니다.

아래 코드와 같이 Viewer의 width, height 값을 지정해야 하며 값을 지정하지 않으면 화면에 Viewer 틀바만 표시됩니다.

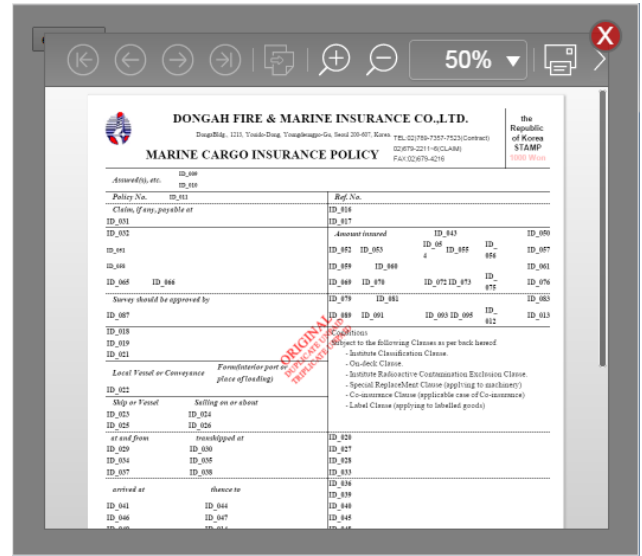
또한 다른 div로 감싸도 되고 위 아래에 다른 엘리먼트가 있어도 됩니다.

Viewer를 브라우저의 전체 화면으로 표시하고 싶은 경우는 body 태그 안에 div 태그만 넣어서 width와 height의 값을 100%로 설정합니다.

```
<div id="crownix-viewer" style="position:absolute; width:700px; height:500px; border:1px solid #000;"></div>
```

오버레이(Overlay)

브라우저 화면 위에 반투명의 레이어가 생성되고 그 위에서 HTML5 Viewer가 실행되는 방식입니다.



<오버레이 방식의 뷰어 실행 전(좌)과 실행 후(우)>

오버레이 방식은 Viewer 영역이 동적으로 생성되기 때문에 뷰어를 위한 div 엘리먼트를 만들 필요가 없습니다. 따라서 뷰어 인스턴스를 생성할 때 뷰어에 대한 div 엘리먼트를 설정하지 않으면 오버레이 방식으로 실행됩니다. 아래 샘플 코드를 참조하십시오.

```

<!DOCTYPE html>
<html style="margin: 0; height: 100%">
<head>
  <meta name="viewport" content="width=device-width, initial-scale=1.0,
    maximum-scale=1.0, minimum-scale=1.0, user-scalable=no" />
  <meta charset="utf-8">
  <script src="../js/jquery-1.11.0.min.js"></script>
  <script src="../js/crownix-viewer.min.js"></script>
  <link rel="stylesheet" type="text/css" href="../css/crownix-viewer.min.css">
  <script type="text/javascript">
    $(document).ready(function(){
      $('#overlay').click(function(){
        // HTML5 Viewer의 인스턴스를 생성하는 코드에서 서버 주소만 지정하고,
        // 뷰어의 id는 지정하지 않았습니다.
        var viewer = new m2soft.crownix.Viewer(serverUrl);
        viewer.open('sample.mrd', '/rfn [sample.txt]');
      });
    });
  </script>
</head>
<body style="margin: 0; height: 100%">
  <input type="button" id="overlay" value="execute" />
</body>
</html>
    
```

Stateful & Stateless 모드

일반적으로 보고서를 실행하면 Reporting Server로부터 전체 페이지를 한번에 다운로드 받습니다.

전체 페이지가 다운로드 되는 과정에서 첫 페이지가 전달되면 화면에 표시합니다.

ü Stateful 모드는?

§ 한 페이지씩 다운로드 받는 방식입니다.

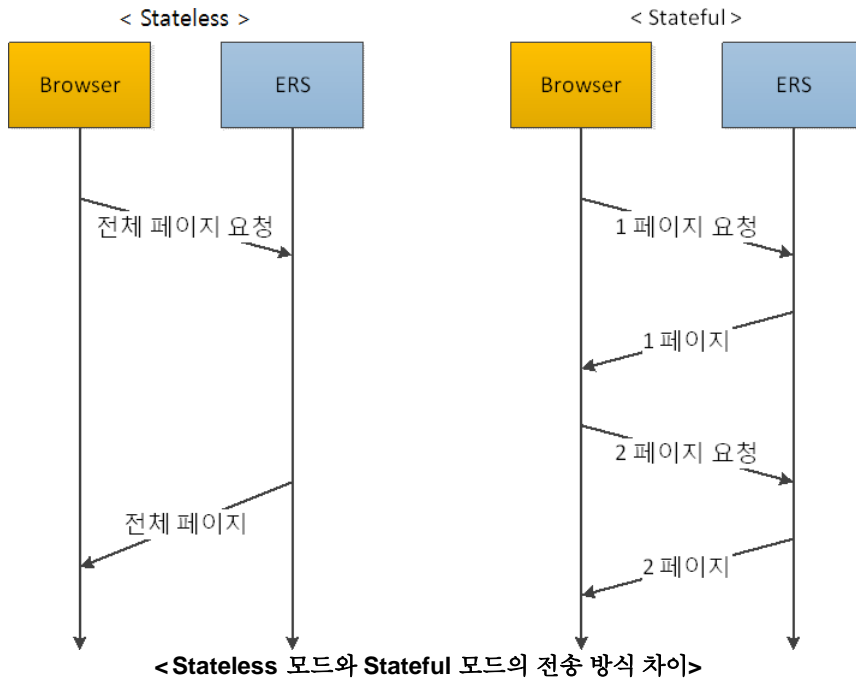
§ 보고서를 실행하면 첫 페이지만 다운로드 받아 화면에 표시하고 사용자가 페이지를 이동하는 시점에 해당 페이지가 서버로 요청되어 다운로드 받는 방식입니다.

ü 네트워크 속도가 느리거나 성능이 낮은 모바일 기기의 사용?

§ 많은 페이지를 받으면서 화면에 표시하거나 페이지를 이동하는 등의 기능 작동이 원활하지 않을 수 있습니다.

§ 사용자가 보고서의 일부 페이지만 보고자 하는 경우 전체를 다운로드 받는 것은 모바일 환경에서 추가 비용이 발생할 수 있습니다.

이와 같은 환경을 고려해서 적절한 **Stateful** 모드를 적용해야 합니다. 아래는 **Stateful** 모드와 일반 모드의 전송 방식의 차이를 나타낸 그림입니다.



Stateful 모드에서 한 번 요청된 페이지는 사용자 기기에 저장되므로 다시 서버로 요청하지 않습니다.

그러므로 이후부터는 페이지가 화면에 빠르게 표시됩니다.

만약 보고서 조회 도중 네트워크가 끊기는 경우 새로운 페이지에 대한 요청 및 표시가 불가능하므로 주의해야 합니다.

HTML5 Viewer의 `setStateLessMode` API 인자로 `false`를 설정하면 **Stateful** 모드로 동작합니다.

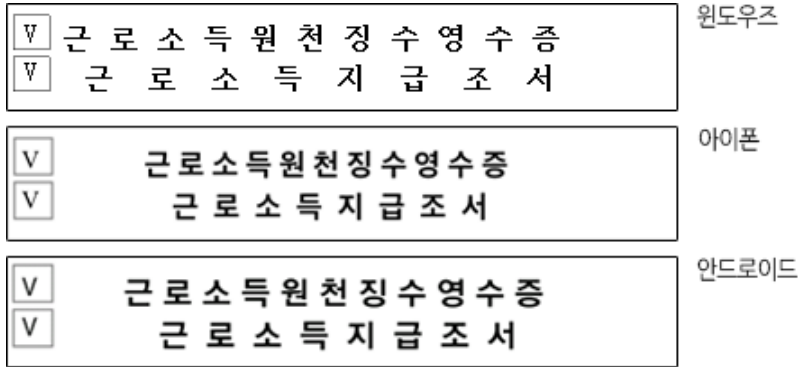
`setStateLessMode` API에 관한 자세한 설명은 [setStateLessMode](#)의 설명을 참조하십시오.

웹 폰트 사용하기

웹 폰트는 폰트 파일을 서버에 올려두고 사용자가 그 폰트 파일을 사용하도록 하는 기술입니다.

운영 체제마다 기본 폰트가 다르기 때문에 웹 폰트를 사용하지 않으면 사용자마다 텍스트의 폰트가 다르게 보일 수 있습니다.

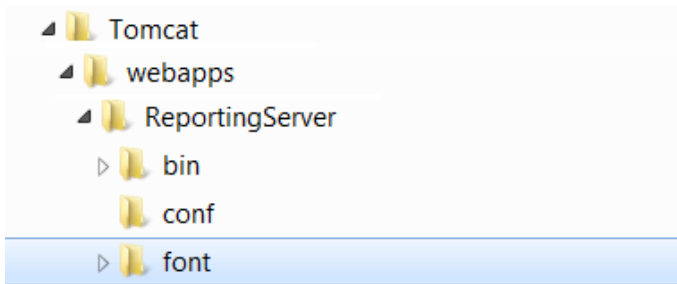
아래 그림을 통해 iOS/Windows/Mac에서 폰트가 다르게 적용되는 것을 살펴볼 수 있습니다.



<운영체제마다 다른 기본 폰트>

웹 폰트는 ttf, woff 확장자를 가지는 폰트들을 지칭하며 브라우저마다 지원하는 확장자가 다릅니다.

웹 폰트를 사용하려면 보고서에서 사용한 폰트들을 Reporting Server의 font 디렉토리에 넣고 `useWebFont` API를 이용하여 웹폰트를 사용하도록 설정해야 합니다.



<Tomcat에서의 폰트 디렉토리 경로 예시>

웹 폰트 설정을 위한 API에 관한 자세한 설명은 [useWebFont](#)의 설명을 참조하십시오.

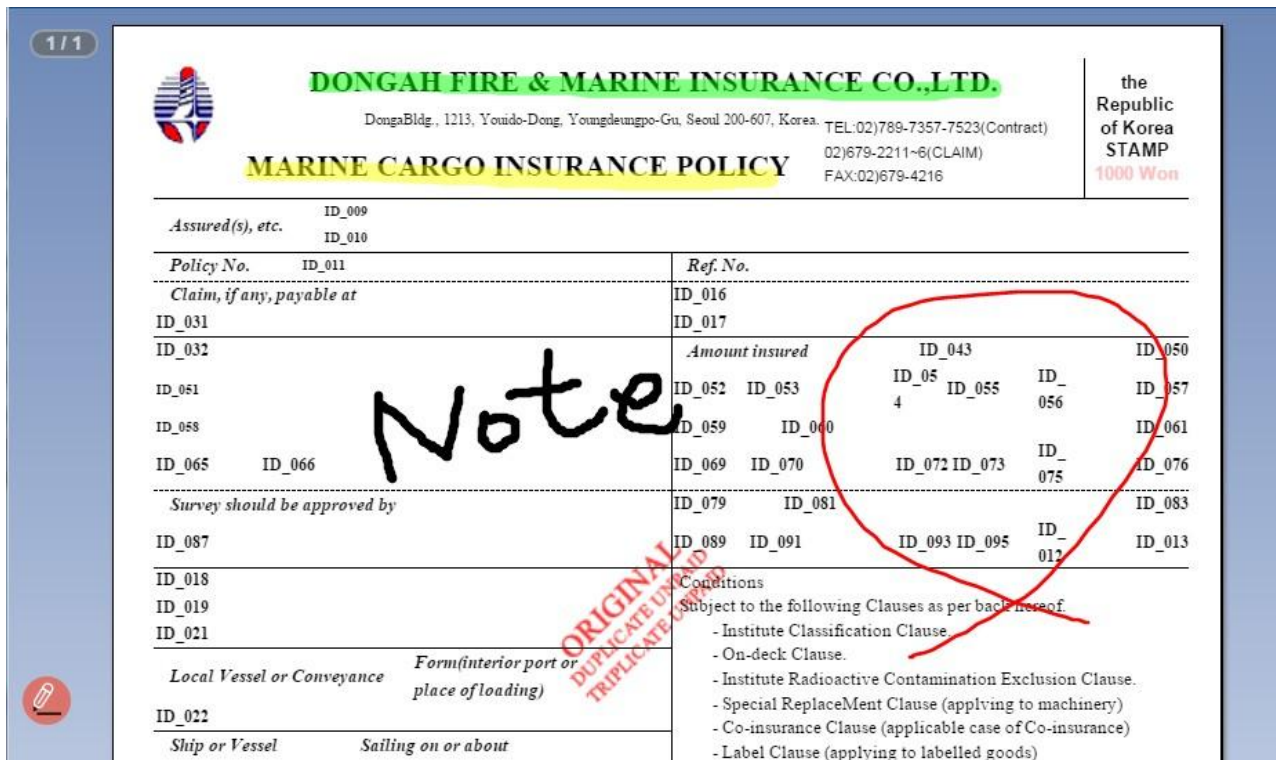
노트 사용하기

노트란 실행된 보고서 영역 내에 자유롭게 메모할 수 있는 기능입니다.

HTML5 Viewer의 [openFile](#) 메서드 호출 시 **enableNote** 속성을 true로 설정하면 노트 모드가 활성화됩니다.

```
var viewer = new m2soft.crownix.Viewer(serverUrl, 'crownix-viewer');
viewer.open('sample.mrd', '/rfn [sample.txt]',
    {enableNote : true}
);
```

위 코드를 실행하면 아래 그림과 같이 마우스나 스크린 터치를 통해 노트를 작성할 수 있습니다.




노트 기능이 활성화 되면 화면의 좌측 하단에 노트 툴바 아이콘이 보이며, 아이콘을 클릭하면 노트 툴바의 아이콘들이 펼쳐집니다.

노트 툴바를 이용하여 펜의 속성(두께, 투명도, 색상)을 변경하거나 지우기, 원래대로, 다시 실행 등의 작업을 할 수 있습니다.

노트 툴바의 각 아이콘 별 기능은 아래와 같습니다.

• HTML5 Viewer의 노트 툴바 아이콘

아이콘	모드	설명
	노트 모드	노트 작성 모드의 아이콘은 별도로 있는 것이 아니며, 마지막으로 선택된 도구(펜, 형광펜, 지우개)의 아이콘이 나타납니다. 최초 선택된 도구는 펜입니다.
	노트 보이기	작성된 노트가 보이지만 편집을 할 수 없는 모드로 전환합니다.
	노트 숨기기	작성된 노트가 보이지 않는 모드로 전환합니다.
	펜	펜을 선택합니다. 펜 상태에서 다시 펜을 클릭하면 도구 속성 대화상자 가 나타납니다. (아래 이미지 참고) 도구 속성 대화상자에서는 두께(1pt ~ 10pt), 투명도(0% ~ 100%), 색상을 설정할 수 있습니다.
	형광펜	형광펜을 선택합니다. 형광펜 상태에서 다시 형광펜 아이콘을 클릭하면 도구 속성 대화상자 가 나타납니다. (아래 이미지 참고) 도구 속성 대화상자에서 두께(5pt ~ 18pt), 색상을 설정할 수 있습니다. 투명도는 50%로 고정입니다.
	텍스트 노트	텍스트 노트를 선택합니다. 텍스트 노트를 선택하고 생성할 위치를 클릭하면 텍스트 노트가 생성 됩니다. 자세한 사항은 텍스트 노트 사용하기 를 참조하세요.
	지우개	지우개를 선택합니다. 지우개 상태에서 노트에 그려진 선을 선택하면 해당 선이 지워집니다. 여러 선이 겹쳐진 부분을 선택한 경우에는 마지막에 그려진 선이 지워집니다. 지우개를 선택하고 텍스트 노트를 선택하면 선택된 텍스트 노트가 지워집니다.
	페이지 노트 삭제	현재 페이지에 작성된 모든 노트를 지웁니다.
	원래 대로	마지막 동작의 직전 상태로 되돌아 갑니다. ※ 최대 100개의 동작까지 되돌릴 수 있습니다. 다른 페이지로 이동했다가 다시 돌아온 경우에는 페이지 이동 전의 작업을 원래대로 되돌릴 수 없습니다.
	다시 실행	원래 대대로 취소한 작업을 다시 실행합니다.

• 도구 속성 대화상자

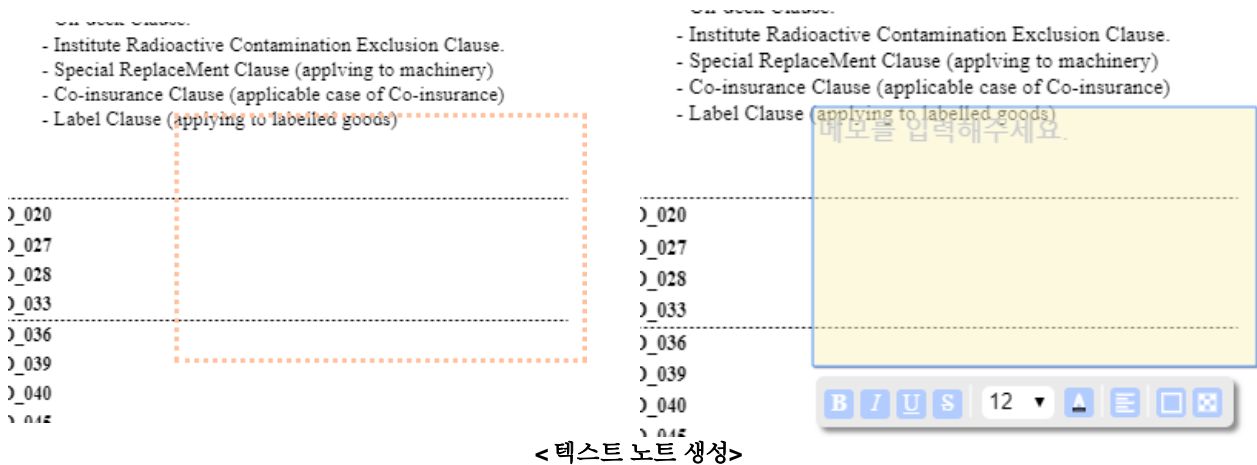
펜, 형광펜의 속성을 변경할 수 있습니다.



텍스트 노트 사용하기

텍스트 노트란 Windows의 스티커 메모 기능처럼 실행된 보고서 영역 내에 자유롭게 메모할 수 있는 기능입니다. 텍스트 노트 버튼을 클릭하고 보고서 영역에서 드래그 앤 드롭으로 영역을 설정하면 그 크기만큼 텍스트 노트가 생성됩니다.

처음 노트가 생성될때는 반투명으로 생성이 되며, 바로 노트를 작성 할 수 있습니다.



• HTML5 Viewer의 텍스트 노트 툴바

아이콘	모드	기본값	설명
	굵게	-	텍스트 노트 글꼴의 굵기를 변경할 수 있습니다. (T/F)
	기울기	-	텍스트 노트 글꼴을 이탤릭체로 변경할 수 있습니다. (T/F)
	취소선 밑줄	-	텍스트 노트 글꼴에 취소선 혹은 밑줄을 넣을 수 있습니다. (두개의 속성 동시 적용 불가)
	폰트 색상	검정색	폰트 색상을 변경할 수 있습니다.
	테두리 선	-	텍스트 노트 테두리 선을 설정할 수 있습니다. (T/F)
	폰트 크기	12 pt	텍스트 노트 글꼴의 크기를 변경할 수 있습니다.
	면 속성	-	텍스트 노트 면 속성을 투명하게하거나 불투명하게 변경할 수 있습니다. (T/F)
	정렬	왼쪽정렬	텍스트 노트 정렬 방식을 변경할 수 있습니다. (왼쪽, 중앙, 오른쪽)

• 도구 속성 대화상자

폰트 색상을 클릭하면 색상 선택 팝업창이 뜨며 원하는 색상을 선택할 수 있습니다.



• 텍스트 노트의 이동과 리사이징

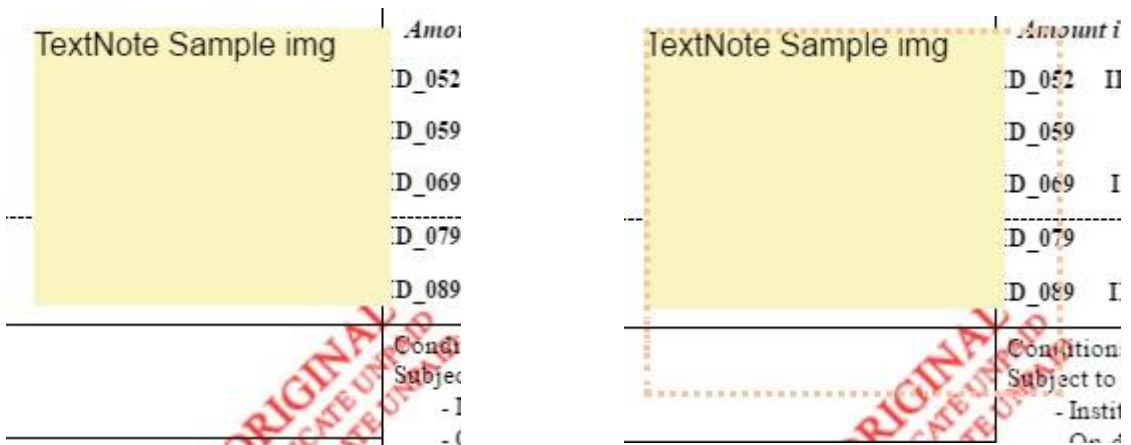
이동과 리사이징은 텍스트 노트 모드 일 때만 가능하며 보고서 영역 밖의 부분으로는 이동 및 리사이징을 할 수 없습니다

ü 이동

텍스트 노트를 클릭하여 드래그를 하면 텍스트 노트의 위치를 이동할 수 있습니다.

ü 리사이징

텍스트 노트 우측 하단 리사이즈 영역을 드래그 하여 텍스트 노트의 크기를 변경할 수 있습니다.



<텍스트 노트 리사이징>

• 셰어링 중 텍스트 노트 기능

셰어링 중 텍스트 노트 기능을 사용할 수 있으며 셰어링하는 모든 사용자들에게 보여집니다.

호스트가 강제 점유를 하고 있을 경우에 클라이언트는 텍스트 노트 기능을 사용할 수 없습니다.

인쇄하기

실행된 보고서를 인쇄하려면 툴바의 인쇄 아이콘 또는 인쇄 메서드를 이용합니다.

보고서 실행 즉시 인쇄를 내보낼 수도 있고 서버에 요청하여 인쇄하는 PDF 인쇄도 수행할 수 있습니다.

PDF 인쇄를 하려면 `print` 메서드에 `option` 파라미터를 전달합니다.

또한 구글 클라우드 프린트 기능을 이용하여 인쇄를 수행 할 수 있습니다.

모바일 기기에서는 Android에서만 인쇄 기능을 지원합니다.

iOS 기반의 모바일 기기에서는 Print 메소드 및 툴바의 인쇄 아이콘이 동작하지 않습니다.

인쇄 API에 관한 자세한 설명은 [print](#)의 설명을 참조하십시오.

클라우드 인쇄 API에 관한 자세한 설명은 [cloudPrint](#)의 설명을 참조하십시오.

조회 툴바 사용하기

조회 툴바 기능이 적용된 보고서를 실행하면 다음 그림과 같이 툴바에 포함된 조회 툴바 메뉴가 활성화됩니다.

The screenshot shows a software interface with a search toolbar at the top. The toolbar contains several icons: a left arrow, a right arrow, a magnifying glass with a plus sign, a magnifying glass with a minus sign, a magnifying glass with a checkmark, and a minus sign. Below the toolbar is a form with multiple sections. The top section has a logo for 'M2 엠투 안과' and a table for 'DNA 검사' with a checked '방상' checkbox. Below that is a table for '김영두 Dr' and '이영두 Dr' with columns for '담당 OP', '자가활형', 'F/U', '입음', 'VTR', '검사번호', and '상당사'. There are several date and time selection tables for '검사 일자', '수술 시간', '통제 시행술', and 'DNA'. The bottom section contains personal information fields like '한글 이름', '영문 이름', '연령', '이메일', '주민번호', '성별', '매입수신', '동의', '지역전화', '휴대폰', '문자 수신', '동학', '주소', '우편번호', '개별', '가족의 녹내장 병력', '복용 약물', '발본 경로', '안타넷 광고', '자연 소개', '업은/지하철 광고', '기타:', and a table for '시력', '교정시력', '자동 굴절 검사', '시각능 검사', '동공 크기 검사', 'TBUT', 'OCT', and '원주각막'.

조회 툴바 메뉴를 선택하면 검색 조건을 입력할 수 있는 대화 상자가 열립니다.

검색 조건을 입력하고 확인 버튼을 클릭하면 선택된 조건을 기준으로 보고서가 재조회됩니다.

검색할 조건을 입력하십시오 (총 3건)

Assured

Policy No.

Date

검색할 조건을 입력하십시오 (총 3건)

Assured

ID_001
 ID_002
 ID_003

Date

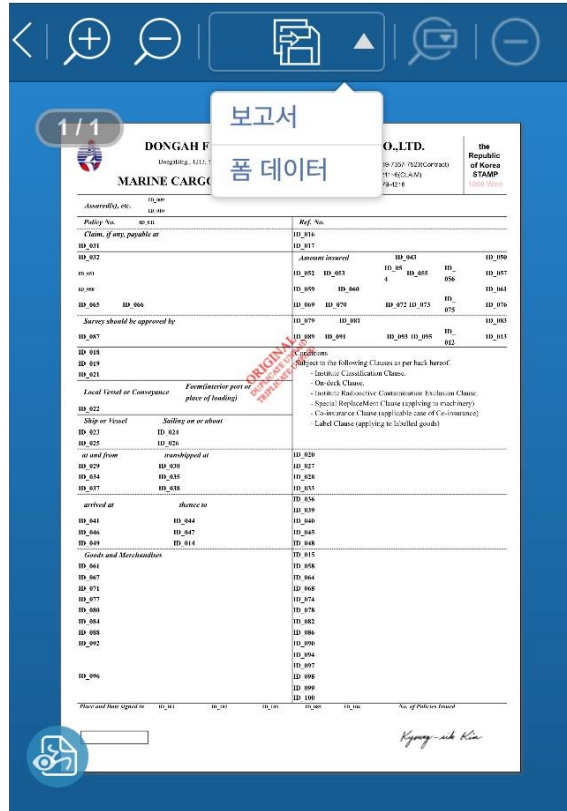
조회된 보고서와 데이터 임시보관

보고서 메뉴를 통하여 현재 출력된 보고서를 임시로 저장하거나 현재 작성중인 폼 데이터만을 임시로 저장 할 수 있습니다.

또한 저장된 데이터를 다시 불러올 수 있습니다.



<폼 편집 모드가 아닐 경우>




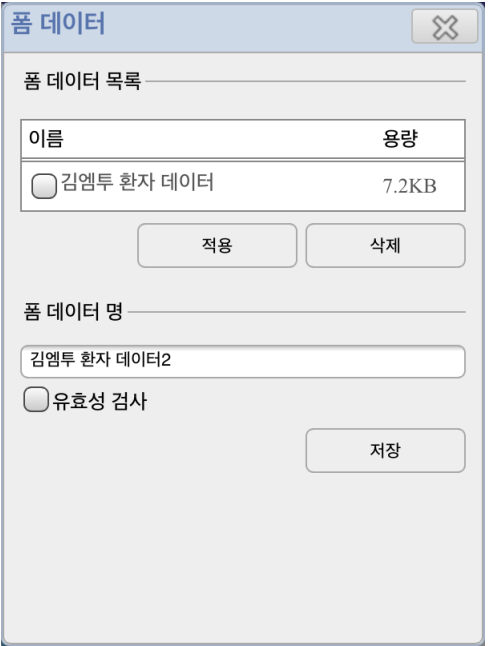
<폼 편집 모드일 경우>

이 기능은 클라이언트에 저장하는 기능입니다.

서버사이드로 구성하려면 viewer에 [infotalkUrl 옵션](#)을 적용해야 합니다. 서버사이드

환경이 설정된 다음 [externalStorage API](#)로 임시보관 합니다.

HTML5 Viewer 모듈 중 보고서 저장 기능이 포함된 js 모듈을 사용하면 위와 같은 메뉴를 사용할 수 있습니다.

메뉴	보고서	폼 데이터
<p>설명</p> <p>보고서 서식을 저장 및 열기를 수행하는 메뉴입니다.</p> <p>보고서 메뉴를 선택하면 저장/열기/삭제를 수행할 수 있는 대화 상자가 열립니다.</p> <p>보고서를 저장할 때 용량을 절약하기 위하여 바코드나 이미지는 제외하고 저장하는 옵션도 선택할 수 있습니다.</p>	<p>폼 편집 모드에서만 수행되며, 폼 편집 중 입력 중인 데이터를 저장하거나, 저장된 데이터를 각각의 폼 필드에 적용하는 기능을 제공합니다.</p> <p>폼 데이터 메뉴를 선택하면 저장/적용/삭제를 수행할 수 있는 대화 상자가 열리며 데이터 저장 전에 필수 데이터를 입력하였는지 제출전과 동일한 유효성 검사를 수행할 수 있는 옵션도 선택할 수 있습니다.</p>	
<p>예시</p>		

데이터를 적용하려면 폼 편집 모드의 보고서가 열려야 데이터를 적용할 수 있습니다.

보고서의 폼과 데이터를 Export하고 Import하기 위한 API에 관한 자세한 설명은 [HTML5 Viewer API 사용법](#)의 [exportMML](#), [exportFormData](#), [importMML](#), [importFormData](#) 메서드의 설명을 참조하십시오.

음성 지원 사용하기

실행된 보고서를 설치 되어있는 센스 리더와 연동하여 음성으로 읽어주는 기능입니다.

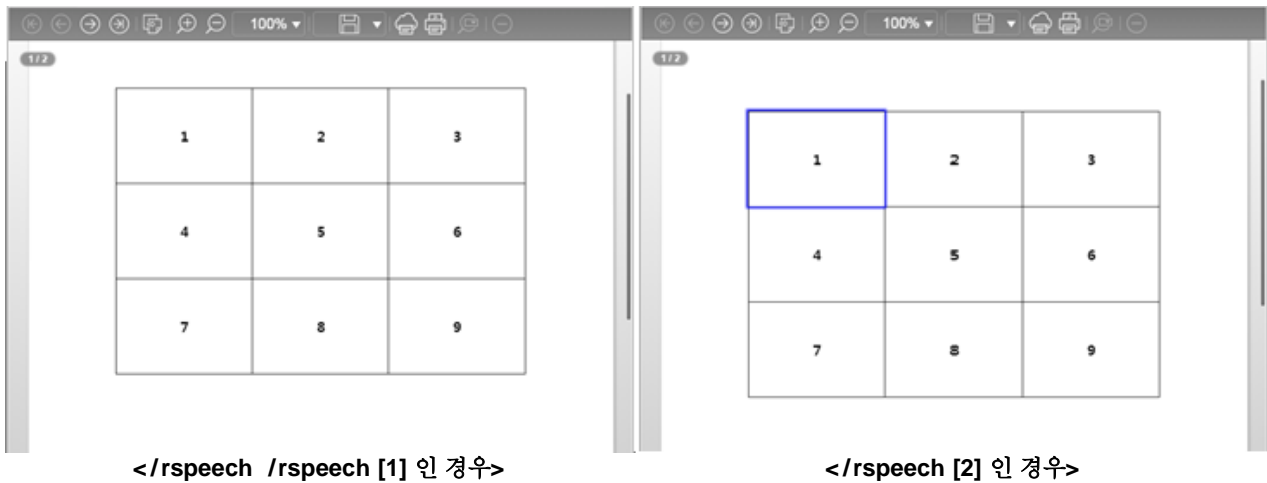
이 기능을 사용하기 위해서는 다음의 파라미터를 사용해야 합니다.

```
/rspeech
/rspeech [1 (기본값) | 2]
```

그리고 센스리더의 가상커서를 켜줍니다.

```
Ctrl + Shift + F12
```

/rspeech [2]를 사용한 경우 이동 시 현재 포커스가 위치한 셀에 파란색 박스가 표시됩니다.



- 이동

ü 셀(글틀)의 이동

실행된 보고서 영역에 있을 때 키보드의 방향키(←,↑,→,↓)를 눌러 셀(글틀) 이동을 할 수 있습니다.

읽는 순서는 보고서를 만들 때 셀이 만들어진 순서에 따라 변경될 수 있으며 표 안의 표로 이동할 수 없습니다.

ü 페이지 이동

실행된 보고서에서 포커스가 마지막 단락에 도착 했을 때 키보드의 PageDown 키를 이용해 다음 페이지 로 이동 할 수 있습니다.

마찬가지로 이전 페이지로 이동하기 위해선 포커스가 가장 첫 번째 단락에 도착 했을 때 키보드의 PageUp 키를 이용해 다음 페이지로 이동 할 수 있습니다.

ü 보고서와 툴바간 이동

보고서와 툴바간 포커스를 이동할 때는 키보드의 Tab키와 Tab + Shift 키를 이용합니다.

§ 보고서 > 툴바 : Tab + Shift 키

§ 툴바 > 보고서 : Tab 키

ü 폼 필드의 이동

폼필드 객체인 경우 셀의 이동과 같이 키보드의 방향키로 이동 할 수 없고 Tab키를 사용해야 합니다.

디자이너에서 탭 인덱스를 지정해야 하며 탭 인덱스 순서대로 이동합니다






● 제약 사항

ü 리더기 설치 여부

본 매뉴얼에서 사용한 리더기는 센스리더 5.6이며 반드시 센스리더 가상 커서 기능을 꺼야합니다.

ü 브라우저 별 음성 지원 가능 여부

모바일 환경은 지원하지 않습니다.

브라우저		지원 버전
	Internet Explorer (Edge 포함)	9.0 이상
	Chrome	54.0 이상
	Safari	X
	FireFox	49.0 이상
	Opera	41.0 이상

PDF로 저장하기

보고서를 PDF로 저장하는 방법에 대하여 설명합니다.

• PDF로 저장하는 방식의 종류

ü 서버 사이트

연결되어 있는 ReportingServer에서 PDF로 저장한 후 다운로드 받는 방식

ü 클라이언트 사이트

HTML5 뷰어에서 직접 PDF로 만들어 Export 하는 방식

- 일반 PDF Export : 폰트 정보를 포함하는 PDF
- 이미지 형태로 Export : 텍스트도 이미지로 처리

• 사용 방법

API와 툴바 사용

ü 서버 사이트 또는 클라이언트 사이트의 방식 선택 : [openFile - clientSidePdf 옵션](#)

ü Export, PDF Print API : [downloadFile\(type\)](#), [export\(type, callback\)](#), [print\(options\)](#)

• 결과 PDF 형태

ü 서버 사이트 & 클라이언트 사이트 일반 PDF 방식 :

폰트 정보가 포함된 파일 형태

ü 클라이언트 사이트 - 이미지 Export 방식 :

PDF가 Base64 형태로 인코딩된 문자열

서버 없이 폰트 포함 PDF 생성하기

클라이언트 사이드로 PDF를 저장하면서 필요한 폰트 정보를 서버에서 가져오는 것이 기본이지만 설정에 따라 서버를 통하지 않을 수 있습니다.






• 동작 원리

PDF에 필요한 폰트 정보는 보고서에서 사용하는 폰트이기 때문에 어떤 폰트를 사용할지 알 수 있습니다. 이미 사용할 폰트 정보를 알 수 있기 때문에 폰트 정보를 js 파일로 작성해 두고 서버 대신 활용합니다.

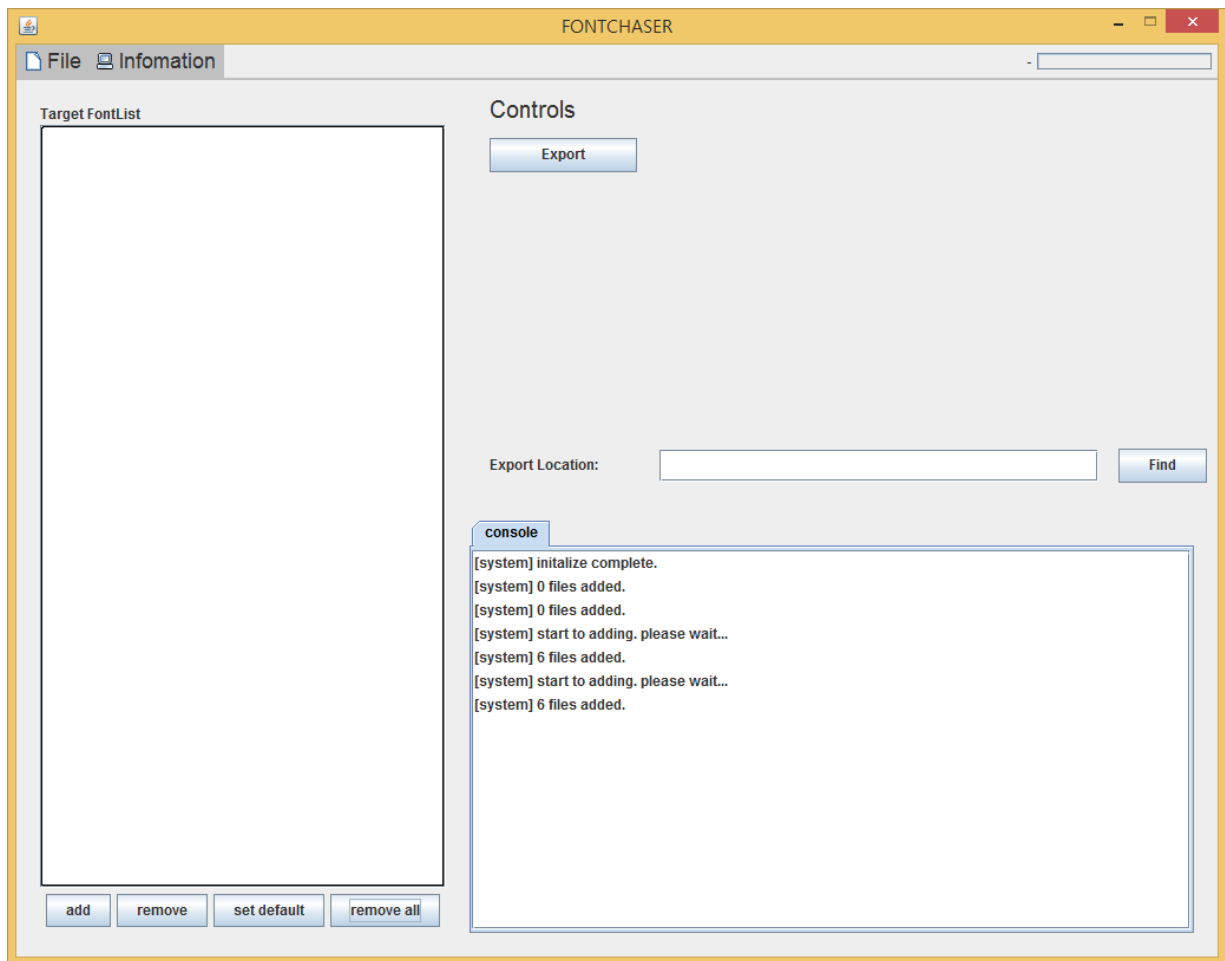
Crownix HTML5 Viewer에서 제공하는 유틸리티 제품인 FontDescriptorGenerator 로 이 js 파일(crownix-fonts.js)을 쉽게 만들 수 있습니다.

• FontDescriptorGenerator로 crownix-fonts.js 만들기

FontDescriptorGenerator는 다음과 같이 구성됩니다.

 lib	- fontDescriptorGenerator.jar 를 동작시키는데 필요한 라이브러리 폴더
 gson-2.2.3.jar	
 iText-4.2.0-m2.jar	
 fontDescriptorGenerator.jar	- 폰트 정보 js 를 작성하는 FontDescriptorGenerator 메인 라이브러리
 run.bat	- fontDescriptorGenerator.jar 실행 파일

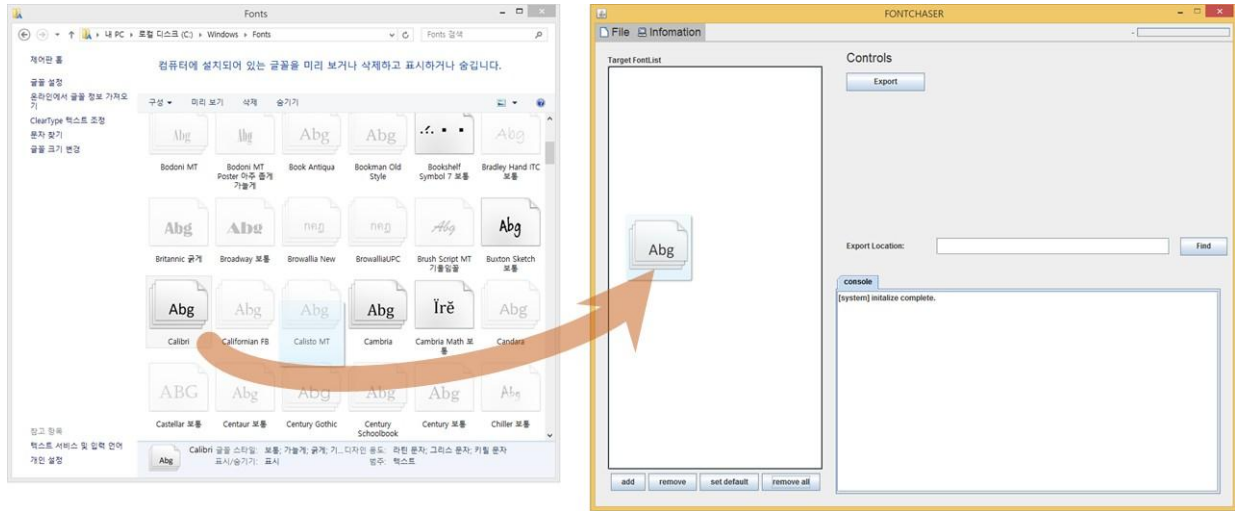
1단계) run.bat으로 FontDescriptorGenerator 실행



2단계) 폰트 추가

폰트를 끌어놓거나 add 버튼으로 추가합니다.

필요 없는 파일이 포함되었다면 remove 버튼으로 적절히 삭제합니다.



ü 기본 폰트 설정하기

추가된 폰트 리스트 중 하나를 기본 폰트로 지정할 수 있습니다.

지정할 폰트를 선택하고 set default 버튼을 눌러 지정하면 폰트 옆에 (default) 표시됩니다.

3단계) 폰트 정보 js 생성

Export Location에서 js를 만들 폴더를 선택하고 Export 버튼을 눌러 crownix-fonts.js 파일을 만듭니다.

● crownix-fonts.js 적용하기

FontDescriptorGenerator로 작성한 crownix-fonts.js를 HTML5 Viewer의 js 라이브러리가 배포된 곳에 배 포함니다.

HTML5 Viewer 소스에 crownix-font.js 링크를 적용하고 클라이언트 사이트 PDF 저장을 합니다.

```

<head>
  <meta charset="utf-8">
  <script src="../js/jquery-1.11.0.min.js"></script>
  <script src="../js/crownix-viewer.min.js"></script>
  <script src="../js/crownix-fonts.js"></script>
  <link rel="stylesheet" type="text/css" href="../css/crownix-viewer.min.css">
</head>
...
    
```

TSA PDF 뷰어 사용하기

PDFJS 기반의 뷰어로 TimeStamp Token으로 만들어진 PDF를 시점확인 (Time Stamp Authority)을 이용해 전자문서의 유효 여부를 확인 할 수 있습니다.

- 사용 방법

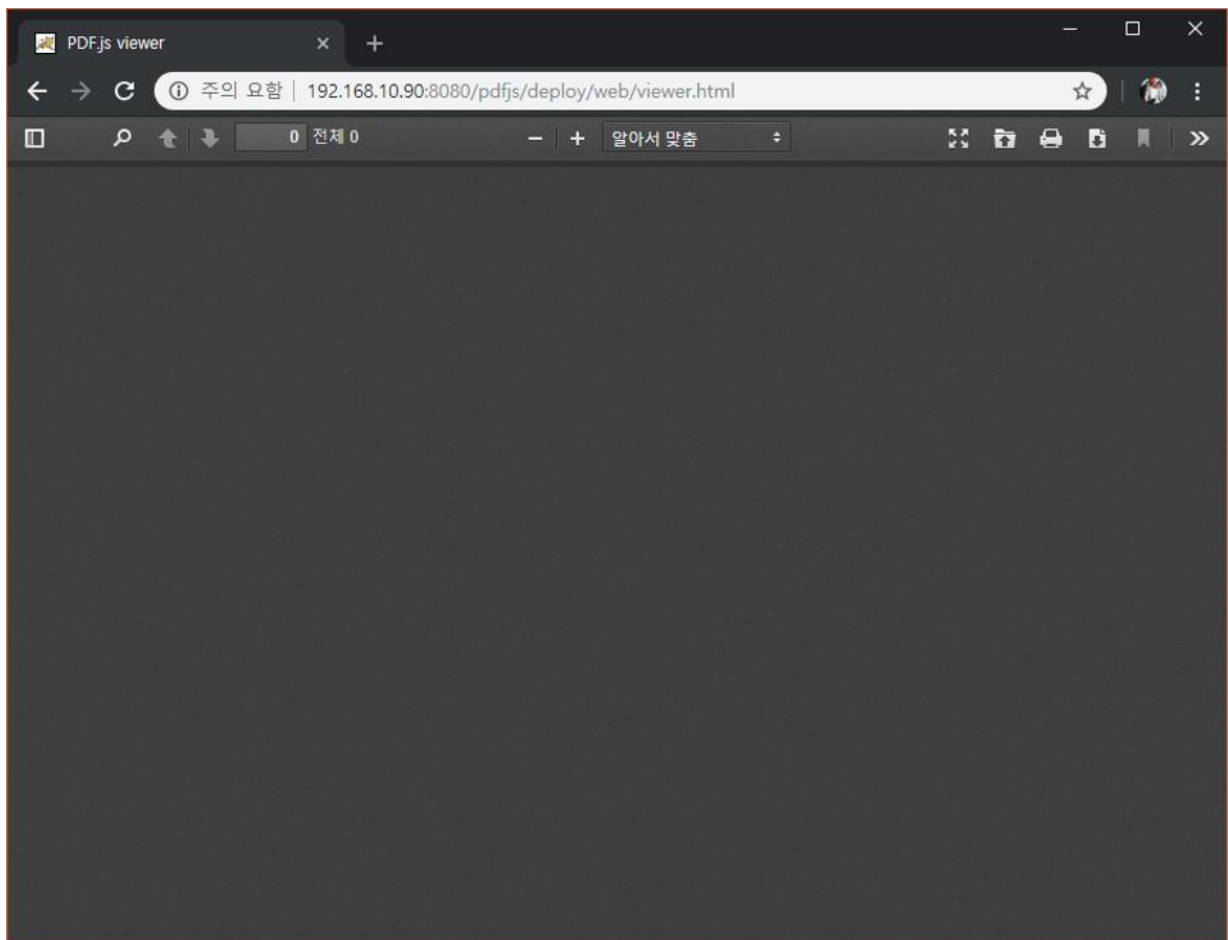
- Time Stamp Token으로 PDF를 생성

- TSA PDF 뷰어 설치

- PDFJS 디렉토리를 WAS_PATH/webapps/ 아래로 이동

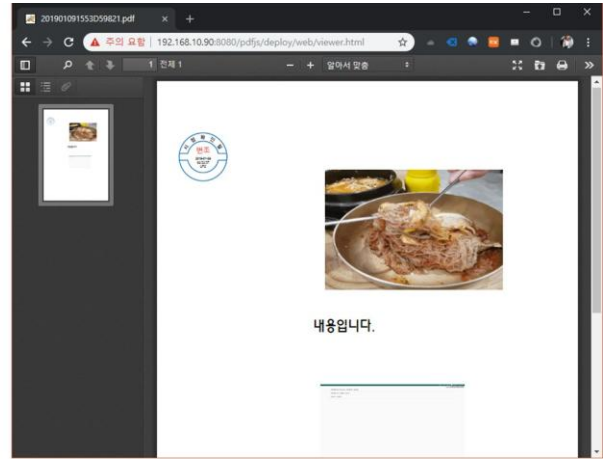
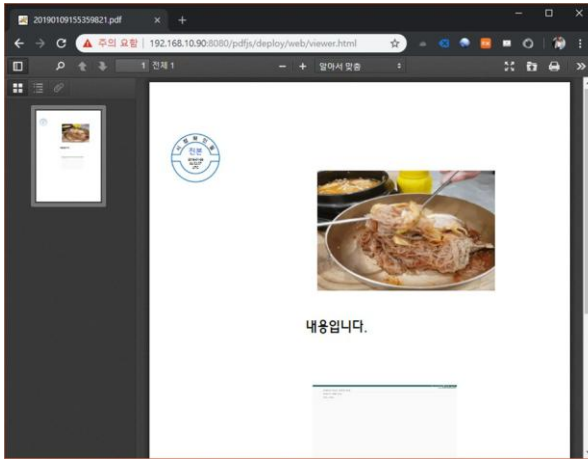
- TSA PDF 뷰어 실행

- 같은 위치에 있는 viewer.html을 호출



- 유효한 PDF와 유효하지 않은 PDF

유효한 PDF의 경우 왼쪽 그림과 같이 진본이라는 스탬프가 찍히고, 유효하지 않은 경우 오른쪽 그림과 같이 변조라는 스탬프가 찍혀서 출력됩니다.



유효한 PDF 스탬프



유효하지 않은 PDF 스탬프

다이얼로그 사용하기

HTML5 Viewer에서 제공하는 API로 HTML5 Viewer의 다이얼로그를 제어하거나 간단한 다이얼로그를 만들 수 있습니다.

• 다이얼로그를 호출하는 API

ü 폼 편집 모드로 실행된 보고서에서 입력 중의 서식(MML) 또는 폼 데이터를 저장하기 위한 다이얼로그 박스 호출

[showMMLStorageDialog\(\)](#), [showFormDataStorageDialog\(\)](#)

ü 조회툴바 호출

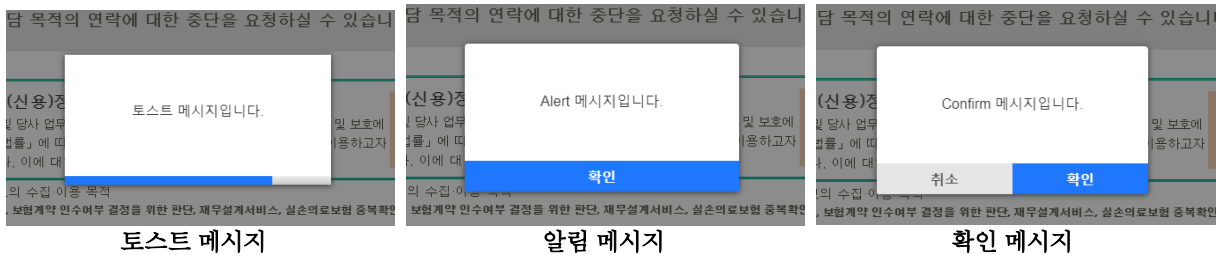
[showInqueryDialog\(\)](#)

ü 페이지 이동 다이얼로그 호출

[showMovePageDialog\(\)](#)

• 다이얼로그를 작성하는 API

ü 다이얼로그 종류(3가지)



ü 이벤트 지원

다이얼로그 박스가 닫히면서 수행할 동작을 정의할 수 있습니다.

다이얼로그 작성을 위한 API에 관한 자세한 설명은 [show\(contents\)](#)의 설명을 참조하십시오.

뷰어 문장 바꾸기

M2soft.crownix.Resource 클래스를 사용해서 기존 문장을 변경할 수 있습니다.

1단계) 리소스 확인하는 코드 작성

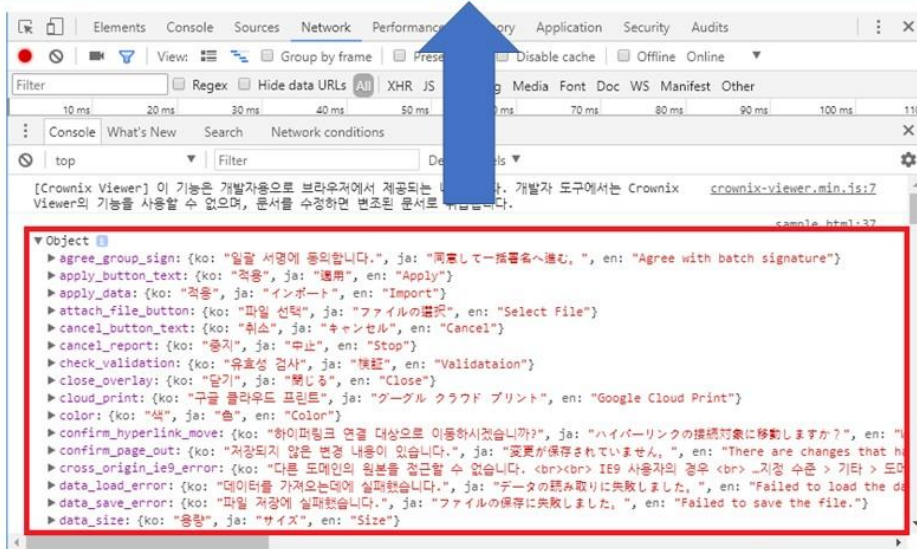
아래와 같이 HTML5 Viewer 스크립트에 문장(리소스)를 확인하는 코드를 작성합니다.

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload =
function(){ var viewer;
var res = m2soft.crownix.Resource.r;
viewer = new m2soft.crownix.Viewer('http://localhost:8080/ReportingServer/service', 'crownix-viewer');
viewer.openFile('sample.mrd', '/rfn [sample.txt]');
console.log(res);
};
</script>
```

2단계) 변경하려는 문장의 키 값 확인



```
Object
▶ agree_group_sign: {ko: "일괄 서명에 동의합니다.", ja: "同意して一括署名へ進む", en: "Agree with batch signature"}
```

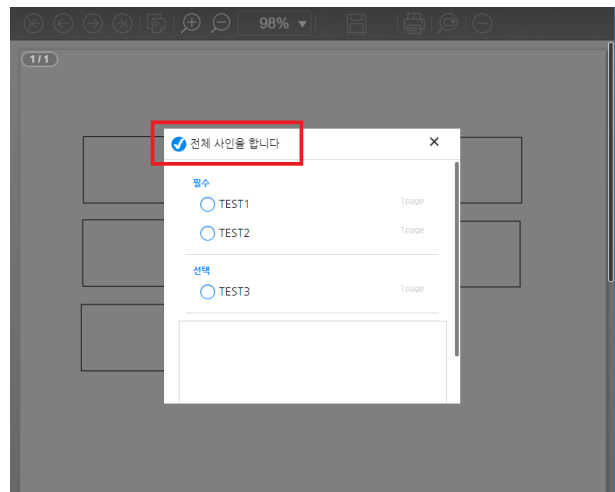
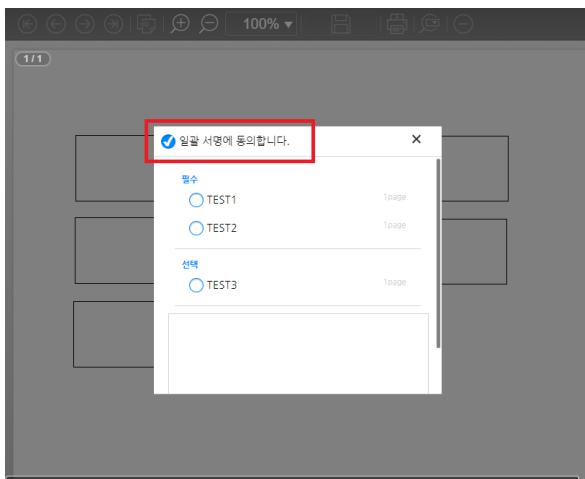


예를 들어, 변경하려는 문장이 "일괄 서명에 동의합니다."일 경우 키값은 "agree_group_sign" 입니다.

agree_group_sign	ko	"일괄 서명에 동의합니다."
키 값	국가 코드	리소스

3단계) 키 값을 확인하고 아래와 같이 문장을 변경합니다.

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload =
function(){ var viewer;
m2soft.crownix.Resource.r['agree_group_sign']['ko'] = "변경하고싶은 내용"; // 한글 문장을 변경
// m2soft.crownix.Resource.r['agree_group_sign']['en'] = "변경하고싶은 내용"; // 영어 문장을 변경
// m2soft.crownix.Resource.r['agree_group_sign']['ja'] = "변경하고싶은 내용"; // 일어 문장을 변경
viewer = new m2soft.crownix.Viewer('http://localhost:8080/ReportingServer/service', 'crownix-viewer');
viewer.openFile('sample.mrd', '/rfn [sample.txt]');
};
</script>
```



데이터 그리드 사용하기

HTML5 Viewer에서 표의 행과 열을 선택하여 표시하거나 숨기는 기능입니다.

Designer의 표 속성에서 '표를 그리드 객체로 사용'을 체크 '데이터 그리드'를 선택하면 HTML5 Viewer의 표를 클릭 시 표의 행과 열을 숨길 수 있는 셀렉트 보드가 표시, 셀렉트 보드를 이용해 행과 열을 선택하여 숨길 수 있습니다.

"높이 지정" 옵션에 대한 설명은 Crownix Report 매뉴얼의 표 속성을 참조하시기 바랍니다.

• 사용 방법

"표를 그리드 객체로 사용"-다이나믹 그리드가 설정된 보고서를 실행시키고 표 클릭 시 자동으로 표시 됩니다.

1 / 1

선택트 보드

+	직원번호	직원이름	직업	관리자번호	입사일	연봉	커미션	직업번호
	7369	SMITH	CLERK	7902	1980-12-17 00:00:00	800		20
	7499	ALLEN	SALESMAN	7698	1981-02-20 00:00:00	1600	300	30
	7521	WARD	SALESMAN	7698	1981-02-22 00:00:00	1250	500	30
	7566	JONES	MANAGER	7839	1981-04-02 00:00:00	2975		20
	7654	MARTIN	SALESMAN	7698	1981-09-28 00:00:00	1250	1400	30
	7698	BLAKE	MANAGER	7839	1981-05-01 00:00:00	2850		30

1 / 1

숨겨진 행/열에 "+" 표시

+	직원번호	직원이름	입사일	연봉	직업번호
	7369	SMITH	1980-12-17 00:00:00	800	20
	7521	WARD	1981-02-22 00:00:00	1250	30
	7654			1250	30
	7369	BLAKE	1981-05-01 00:00:00	2850	30
	7782	CLARK	1981-06-09 00:00:00	2450	10
	7788	SCOTT	1987-04-19 00:00:00	3000	20

기존 표에서 숨겨진 행/열 만큼 표가 축소

숨기고 싶은 행/열을 클릭 후 우측 버튼 클릭 시 "행 숨기기" 표시

행 숨기기

보고서에 PDF를 삽입하여 사용하기

HTML5 Viewer에서 PDF파일을 삽입해 여는 기능입니다.

OpenFile에서 바로 PDF를 열거나 Designer에서 서브페이지를 PDF로 설정하여 열 수 있습니다. (단, Win32 Engine은 동작하지 않고 HTML5 Viewer 옵션인 [textOnCanvas](#)가 기본값으로 설정되어 동작 합니다.)

• 사용 방법

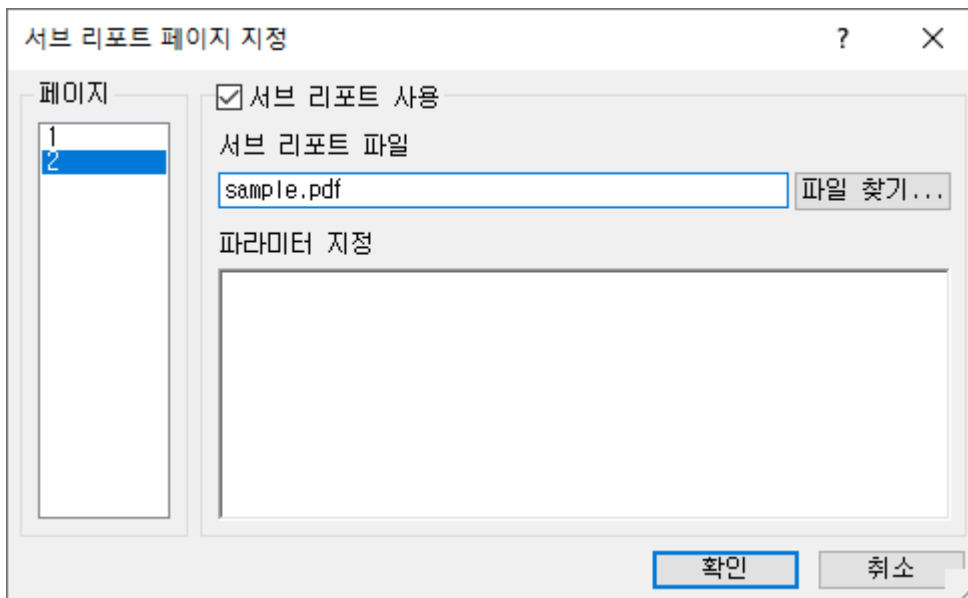
▣ openFile에 PDF를 직접 넣어 사용하는 방법

openFile에 직접 PDF 파일을 지정하여 사용할 수 있으며 Append Report로도 사용할 수 있습니다.

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload =
function(){ var viewer;
viewer = new m2soft.crownix.Viewer('http://localhost:8080/ReportingServer/service', 'crownix-viewer');
viewer.openFile('sample.pdf', '/rpdfimportinfo [U, 3, 10, 1234]'); // 1234로 암호가 걸린 PDF를 3 page부터 10pa
};
</script>
```

▣ 서브 페이지에 PDF를 설정하여 사용하는 방법

Designer에서 서브 리포트 페이지 지정에 PDF파일을 설정하여 PDF를 삽입합니다.



/rpdfimportinfo 파라미터에 대한 자세한 설명은 Designer 매뉴얼을 참조하십시오.

세로 스크롤 사용하기

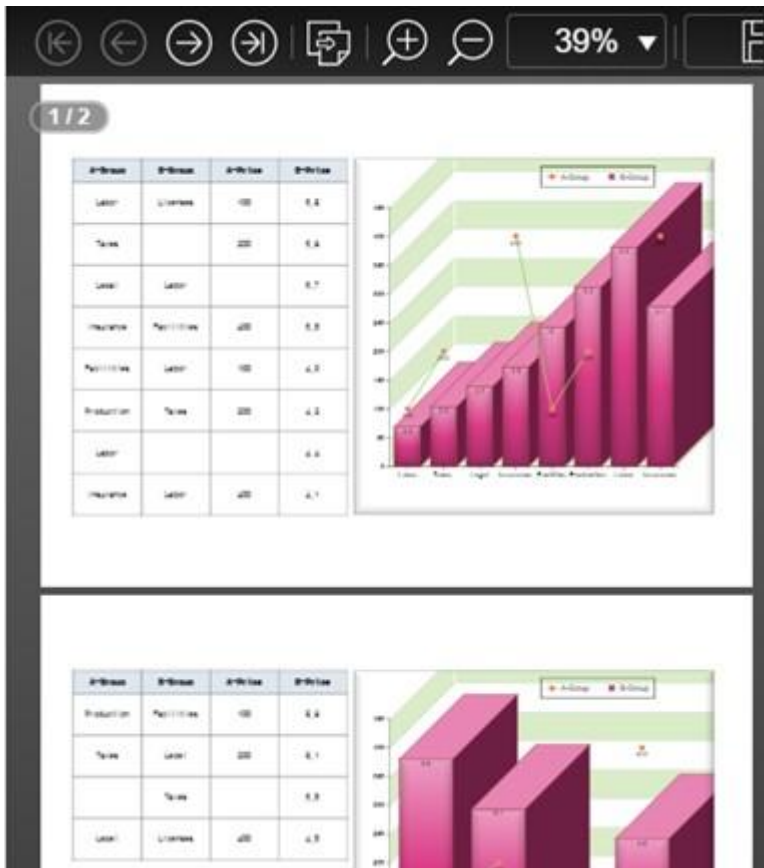
페이지를 넘기지 않고 여러장의 보고서를 스크롤로 내려서 볼수 있는 기능입니다.

(단, 스마트 폼 웨어링, 미리링 기능과 사용할 수 없고, HTML5 Viewer 옵션인 [textOnCanvas](#)가 기본값으로 설정되어 동작 합니다).

세로 스크롤을 사용하기 위한 API에 관한 자세한 설명은 [scrollOptions](#)의 useInfiniteScroll 설명을 참조하십시오.

• 사용 방법

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload =
function(){ var viewer;
viewer = new m2soft.crownix.Viewer('http://localhost:8080/ReportingServer/service', 'crownix-viewer');
viewer.openFile('sample.mrd', '/rfn [sample.txt]', {scrollOptions : {useInfiniteScroll : true }});
};
</script>
```



HTML5 Viewer API 사용법

이 섹션에서는 HTML5 Viewer의 API를 설명합니다.

폼 필드와 관련된 API는 Smart Form 라이선스가 적용된 제품에서만 사용할 수 있습니다.

M2soft.crownix.Viewer의 API

M2soft.crownix.Viewer 클래스에서 제공하는 API에 대하여 설명합니다. 포함된

API들은 뷰어의 인스턴스를 정의하고 이를 통하여 이용할 수 있습니다.

ü API는 openFile API를 사용한 다음 그 외의 API를 작성하는 것을 권장하며 대소문자를 구분합니다.

viewer()

정의

m2soft.crownix.Viewer(string serverPath, string bindingID)

m2soft.crownix.Viewer(string serverPath, object options)

설명

뷰어 생성자입니다.

인수

serverPath : *string*

보고서를 생성에 필요한 Reporting Server의 접속 URL을 설정

bindingID : *string*

일반 모드에서 뷰어를 바인딩 할 div의 ID를 지정

options : *object*

Viewer의 생성자 옵션

설정값 : { viewerWidth, viewerHeight, enableCloseButton, toolbarScale, infotalkUrl, share }

viewerWidth : *number*

overlay 모드에서 팝업창의 가로 크기

설정값 : 0~1 사이의 소수 (1 : 100% 를 의미함)

기본값 : 0.9

viewerHeight : *number*

overlay 모드에서 팝업창의 세로 크기

설정값 : 0~1 사이의 소수 (1 : 100% 를 의미함)

기본값 : 0.9

enableCloseButton : *boolean*

overlay 모드에서 오른쪽 끝의 x 버튼 표시 유무 결정

설정값 : true, false (기본값)

toolbarScale : *number*

설정값 : 0.5~2 사이의 소수 (1 : 100% 를 의미함)

기본값 : 1

toolbarScale 옵션은 모바일에서 사용하는 기능입니다. 데스크탑에서 사용하면 정상적으로 사용되지 않습니다.

infotalkUrl : *string*

서버사이드 중간 저장을 위해 사용할 InfoTalk Server의 주소

share : *object*

Smart Form Sharing을 사용하기 위한 옵션

예

ü DIV 에 바인딩해서 실행하는 경우

```
<!DOCTYPE html>
<html style="margin: 0; height: 100%">
<head>
  <meta charset="utf-8">
  <script src="../../js/jquery-1.11.0.min.js"></script>
  <script src="../../js/crownix-viewer.min.js"></script>
  <link rel="stylesheet" type="text/css" href="../../css/crownix-viewer.min.css">
</head>
<body style="margin: 0; height: 100%">
  <div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
  <script>
    window.onload = function(){
      var viewer = new
        m2soft.crownix.Viewer( 'http://localhost:8080/ReportingServer/service', 'crownix-viewer');
      viewer.openFile('sample.mrd', '/rfrn [sample.txt]');
    };
  </script>
</body>
</html>
```

Overlay로 띄우면서 꼭찬 화면으로 실행하는 경우

아래 옵션을 사용하지 않고 `serverPath` 한개만 파라미터로 지정하면 `overlay` 모드로 실행되지만 꼭찬 화면으로 나오지는 않습니다.

```
<!DOCTYPE html>
<html style="margin: 0; height: 100%">
<head>
  <meta charset="utf-8">
  <script src="../../js/jquery-1.11.0.min.js"></script>
  <script src="../../js/crownix-viewer.min.js"></script>
  <link rel="stylesheet" type="text/css" href="../../css/crownix-viewer.min.css">
</head>
<body style="margin: 0; height: 100%">
  <script>
    window.onload = function(){
      $('#overlay1').click(function(){
        var viewer = new m2soft.crownix.Viewer(
          'http://localhost:8080/ReportingServer/service',
          {viewerWidth: 1.0, viewerHeight: 1.0, enableCloseButton: false});
        viewer.openFile('sample.mrd', '/rfn [sample.txt]');
      });
    };
  </script>
  <input type="button" id="overlay1" value="execute" />
</body>
</html>
```

툴바 스케일 조절하기

```
<!DOCTYPE html>
<html style="margin: 0; height: 100%">
<head>
  <meta charset="utf-8">
  <script src="../../js/jquery-1.11.0.min.js"></script>
  <script src="../../js/crownix-viewer.min.js"></script>
  <link rel="stylesheet" type="text/css" href="../../css/crownix-viewer.min.css">
</head>
<body style="margin: 0; height: 100%">
  <script>
    window.onload = function(){
      $('#overlay1').click(function(){
        var viewer = new m2soft.crownix.Viewer(
          'http://localhost:8080/ReportingServer/service',
          {toolbarScale: 2});
        viewer.openFile('sample.mrd', '/rfn [sample.txt]');
      });
    };
  </script>
  <input type="button" id="overlay1" value="execute" />
</body>
</html>
```

DONGAH FIRE & MARINE INSURANCE CO.,LTD.
 DongaBldg., 1213, Youido-Dong, Youngdeungpo-Gu, Seoul 200-607, Korea. TEL:02)789-7357-7523(Contract)
MARINE CARGO INSURANCE POLICY 02)679-2211-6(CLAIM) FAX:02)679-4216

the Republic of Korea STAMP 1000 Won

<i>Assured(s), etc.</i>	ID_009 ID_010			
<i>Policy No.</i>	ID_011	<i>Ref. No.</i>	ID_016	
<i>Claim, if any, payable at</i>	ID_031 ID_032	ID_017		
		<i>Amount insured</i>	ID_043	ID_050

share 옵션

정의

```
m2soft.crownix.Viewer(string serverPath, {
    share : { url, mmlFromServer, reconnection, reconnection }
})
```

설명

스마트 폼 웨어링을 사용하기 위한 share 옵션이며 `socket.io.js` 파일을 추가해야 합니다.

설정 값

url : string

Share Server url과 포트번호 (필수)

config.json 파일에서 포트번호를 변경할 수 있습니다.

mmlFromServer : boolean

Share Server에 저장된 정보

설정값 :

true : MML 정보를 가져옵니다.

false : MRD 정보(path, param, option)만 가져와서 client가 직접 MML을 요청합니다.

reconnection : boolean

재연결 시도 여부

설정값 :

true : 재 연결 시도를 허용합니다.

false : 재 연결 시도를 허용하지 않습니다.

timeout : number

서버와의 timeout 시간 설정 (초 단위)

예

```
<!DOCTYPE html>
<html style="margin: 0; height: 100%">
<head>
  <meta charset="utf-8">
  <script src="../js/jquery-1.11.0.min.js"></script>
  <script src="../js/crownix-viewer.min.js"></script>
  <script src="../js/share/node_modules/socket.io/node_modules/
    socket.io-client/socket.io.js"></script>
  <link rel="stylesheet" type="text/css" href="../css/crownix-viewer.min.css">
</head>
<body style="margin: 0; height: 100%">
  <script>
    window.onload = function(){
      var viewer = new m2soft.crownix.Viewer(
        'http://localhost:8080/ReportingServer/service',
        { share : { url : 'ServerURL:52280'}});

      viewer.openFile('sample.mrd', '/rfrn [sample.txt]');
    };
  </script>
</body>
</html>
```


setRData(dataset)

정의

setRData(string dataset)

설명

FILE 접속방식 또는 XMLFILE 접속방식 보고서에 사용하는 메서드로 보고서 실행에 필요한 데이터 셋을 지정합니다.

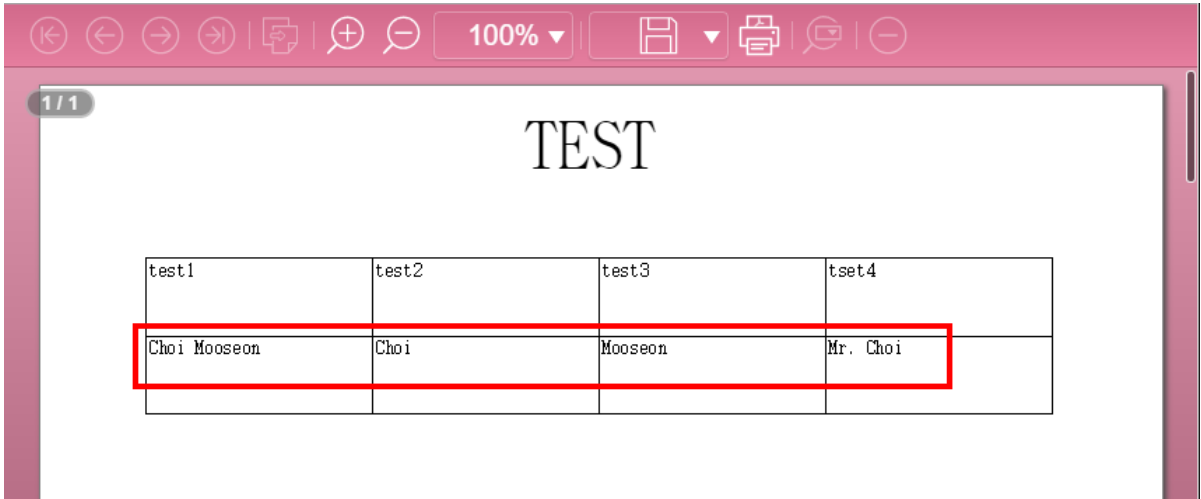
보고서에서 만들었던 필드명과 필드 개수가 같아야 합니다.

예

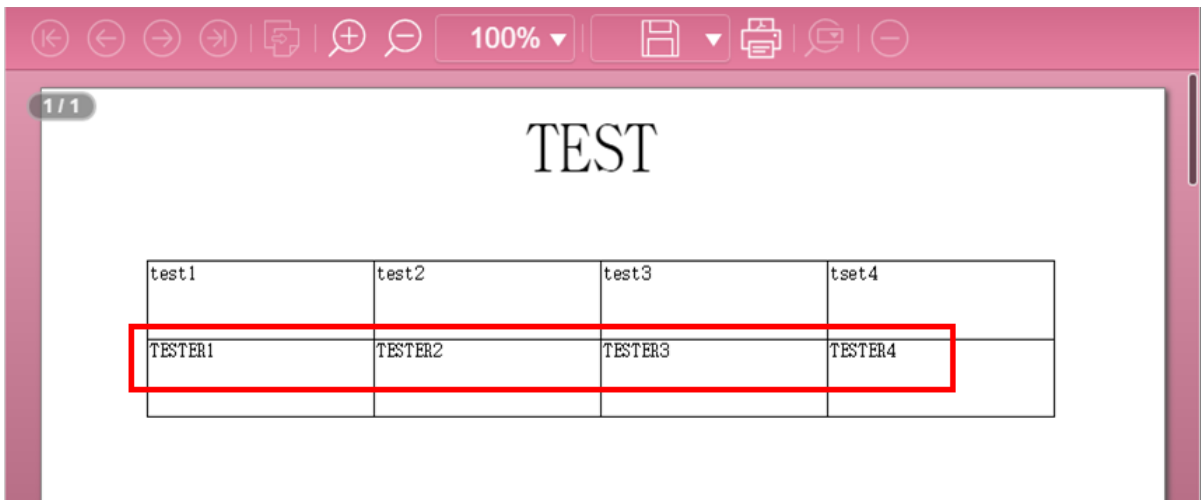
FILE 접속방식

```
viewer.setRData("TESTER1,TESTER2,TESTER3,TESTER4");
```

setRData를 사용하지 않았을 경우는 다음과 같이 보고서에 설정된 데이터가 나타납니다.



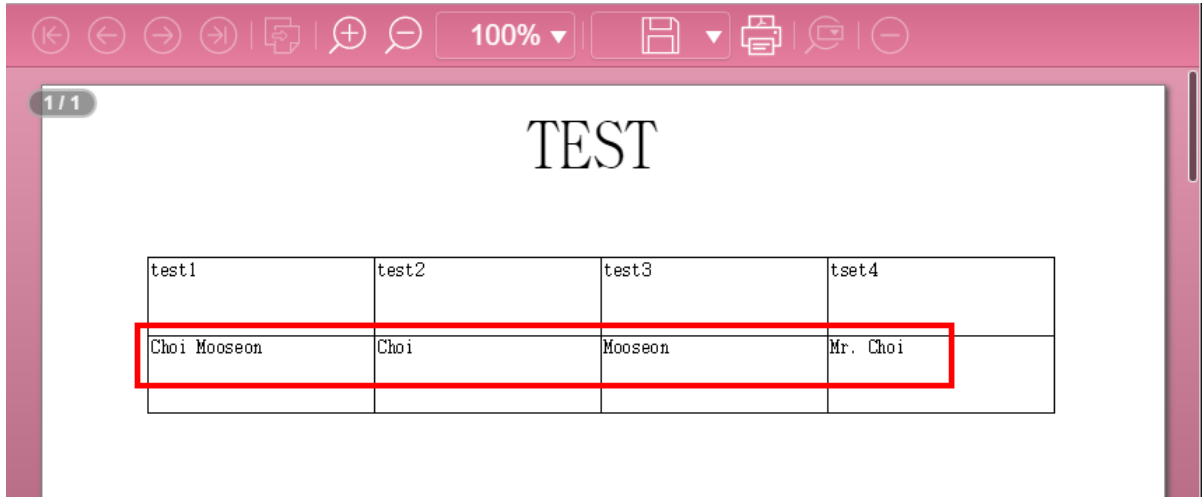
setRData를 사용했을 경우는 다음과 같습니다.



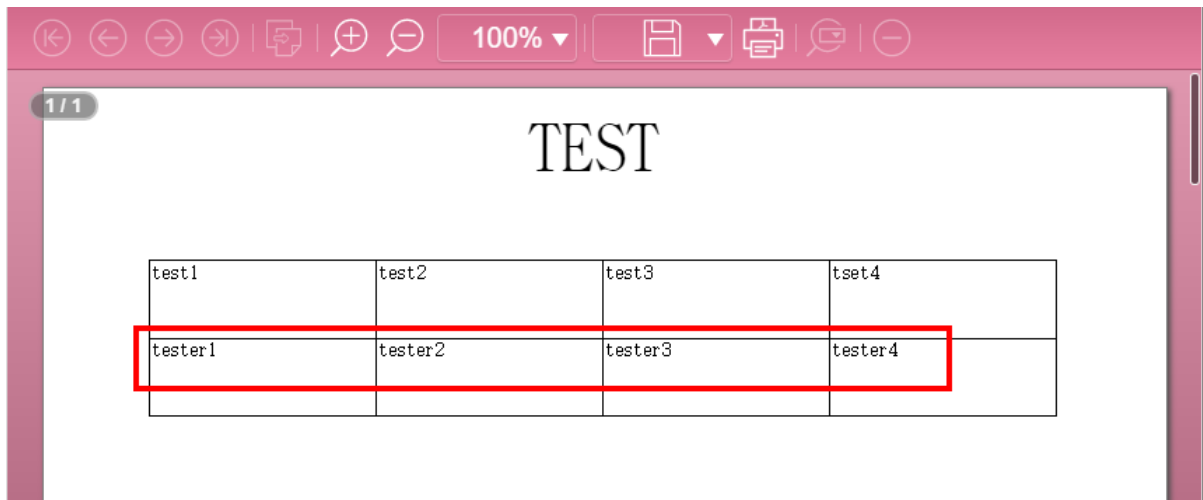
ü XMLFILE 접속방식

```
var xml = "<?xml version='1.0' encoding='utf-8'?><root><test><t1>tester1</t1>"
xml = xml + "<t2>tester2</t2><t3>tester3</t3><t4>tester4</t4></test></root>";
viewer.setRData(xml);
```

setRData를 사용하지 않았을 경우는 다음과 같이 보고서에 설정된 데이터가 나타납니다.



setRData를 사용했을 경우는 다음과 같습니다.



setParameterEncrypt(opt)

정의

setParameterEncrypt(short opt)

설명

FileOpen 메서드를 호출할 때 파라미터를 암호화해서 넘길지 여부를 지정합니다.

HTML5 뷰어에서 소스보기를 했을 때 데이터가 암호화된 것을 확인할 수 있습니다.

html 파일로 저장하는 것이 아니라 jsp 파일로 저장해야 합니다

인수

opt : number

암호화 방식.

openFile API 실행 시 암호화된 파라미터를 넘기려면 설정 값을 입력합니다.

설정 값 : 0~3 사이의 정수

예

```
<%@ page import="java.io.*,java.util.*,enc.*" %>
<%
C.setCharset("UTF- 8");

int type =3;

String mrd_filepath = "sample.mrd";
String mrd_path = new String(C.process(mrd_filepath,type));

String param = "/ rxmlreportopt [ 0 ] ";
String mrd_param = new String(C.process(param,type));

String mrd_data = "<?xml version=' 1.0' encoding='utf- 8' ?
><root><test><t1>tester1</ t1><t2>tester2</ t2><t3>tester3</ t3><t4>tester4</ t4></ test></ root>";
String rdata = new String(C.process(mrd_data,type));

%>
<!DOCTYPE html>
<html style="margin:0;height:100%">
  <head>
    <meta charset="utf- 8">
  ...
</script>
window.onload = function(){
  m2soft.crownix.Viewer.debug = false ;
  var viewer = new m2soft.crownix.Viewer(
'http:// localhost:8080/ ReportingServer/ service', 'crownix- viewer' );
viewer.setParameterEncrypt(<%=type%>);
viewer.setRData("<%=rdata%>");
```

```

viewer.openFile('<%=mrd_path%>', '<%=mrd_param%>');
};
</script>
</body>
</html>

```

페이지 소스 보기 결과

setParameterEncrypt을 사용하지 않고 페이지 소스보기를 했을 경우

```

1 <!DOCTYPE html>
2 <html style="margin:0;height:100%">
3   <head>
4     <meta charset="utf-8">
5     <title>Crownix HTML5 Viewer</title>
6     <script src=".../js/jquery-1.11.0.min.js"></script>
7     <script src=".../js/crownix-viewer.min.js"></script>
8     <link rel="stylesheet" type="text/css" href=".../css/crownix-viewer.min.css">
9
10  </head>
11  <body style="margin:0;height:100%">
12    <div id="crownix-viewer" style="position:absolute;width:100%;height:100%"></div>
13    <script>
14      window.onload = function(){
15        m2soft.crownix.Layout.setTheme('pink');
16        var viewer = new m2soft.crownix.Viewer('http://localhost:8080/ReportingServer/service', 'crownix-viewer');
17        viewer.setData("<?xml version='1.0' encoding='utf-8'?><root><test><t1><t2>tester1</t1><t2>tester2</t2><t3>tester3</t3><t4>tester4</t4></test></root>");
18        viewer.openFile('test.mrd');
19      };
20    </script>
21  </body>
22 </html>

```

setParameterEncrypt 사용했을 경우

```

1
2
3
4
5 <!DOCTYPE html>
6 <html style="margin:0;height:100%">
7   <head>
8     <meta charset="utf-8">
9     <title>Crownix HTML5 Viewer</title>
10    <script src="http://localhost:8080/ReportingServer/sample/js/jquery-1.11.0.min.js"></script>
11    <script src="http://localhost:8080/ReportingServer/sample/js/crownix-viewer.min.js"></script>
12    <link rel="stylesheet" type="text/css" href="http://localhost:8080/ReportingServer/sample/css/crownix-viewer.min.css">
13
14  </head>
15  <body style="margin:0;height:100%">
16    <div id="crownix-viewer" style="position:absolute;width:100%;height:100%"></div>
17    <script>
18      window.onload = function(){
19        //m2soft.crownix.Viewer.debug = false;
20        m2soft.crownix.Layout.setTheme('pink');
21        var viewer = new m2soft.crownix.Viewer('http://localhost:8080/ReportingServer/service', 'crownix-viewer');
22
23        /* Encrypt */
24        viewer.setParameterEncrypt(3);
25
26        viewer.setData("<?xml version='1.0' encoding='utf-8'?><root><test><t1><t2>tester1</t1><t2>tester2</t2><t3>tester3</t3><t4>tester4</t4></test></root>");
27        viewer.openFile("<?xml version='1.0' encoding='utf-8'?><root><test><t1><t2>tester1</t1><t2>tester2</t2><t3>tester3</t3><t4>tester4</t4></test></root>");
28      };
29    </script>
30  </body>
31 </html>

```

openFile

보고서를 실행할 때 사용하는 API를 제공합니다.

[openFile\(mrdPath, mrdParam, options\)](#)

[openFile\(mrdList, object options\)](#)

[options](#)

openFile(mrdPath, mrdParam, options)

정의

```
openFile(string mrdPath, string mrdParam)
```

```
openFile(string mrdPath, string mrdParam, object options)
```

설명

1개의 보고서를 실행할 때 사용합니다.

인수

mrdPath : string

실행할 mrd 파일의 경로입니다. 상대 경로와 URL 경로를 지원합니다.

상대 경로의 경우 Reporting Server의 mrd 경로를 기준으로 한 상대 경로입니다. URL 경로의 경우 서버에서 mrd를 다운로드 받아서 실행하므로 속도가 저하될 수 있습니다. 따라서, 상대 경로를 사용하는 것을 권장합니다.

mrdParam : string

mrd 실행에 필요한 파라미터를 설정합니다.

파라미터에 대한 상세한 설명은 Crownix Report 파라미터 매뉴얼을 참조하시기 바랍니다.

options : object

Viewer의 실행 옵션입니다.

[options](#) 의 내용을 참고하세요

예

```
viewer.openFile('sample1.mrd', '/rfn [sample2.txt]');
viewer.openFile('sample1.mrd', '/rfn [sample2.txt]', options)
```

openFile(mrdList, options)

정의

openFile(array mrdList)

openFile(array mrdList, object options)

설명

여러 개의 보고서를 하나의 보고서로 연결하여 실행할 때 사용합니다.

인수

mrdList : array

연결할 mrd를 리스트 형태로 지정합니다.

mrd는 파일 경로이며, 상대 경로와 URL 경로를 지원합니다.

포함되는 속성은 [openFile\(mrdPath, mrdParam, options\)](#)의 mrdpath와 mrdParam의 정의와 동일합니다.

options : object

Viewer의 실행 옵션입니다.

[options](#)의 내용을 참고하세요.

예

여러 개의 보고서를 붙여 실행할 경우는 다음과 같이 사용합니다.

```
var mrdList =
  [{mrdPath:'sample1.mrd', mrdParam :'/rfn [sample1.txt]},
   {mrdPath:'sample2.mrd', mrdParam :'/rfn [sample2.txt]}];
viewer.openFile(mrdList);
```


options

openFile API를 사용할 때 옵션을 사용할 수 있습니다.

```

<!DOCTYPE html>
<html style="margin: 0; height: 100%">
<head>
  <meta charset="utf-8">
  <script src="../../js/jquery-1.11.0.min.js"></script>
  <script src="../../js/crownix-viewer.min.js"></script>
  <link rel="stylesheet" type="text/css" href="../../css/crownix-viewer.min.css">
</head>
<body style="margin: 0; height: 100%">
  <div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
  <script>
    window.onload = function(){
      var viewer = new
      m2soft.crownix.Viewer( 'http://localhost:8080/R
      eportingServer/service', 'crownix-viewer');
      viewer.openFile('sample.mrd', '/r/n [sample.txt]');
    };
  </script>
</body>
</html>

```

위 코드에서 정의된 옵션을 사용하려면 다음과 같이 변경합니다.

```

var options = { //속성 설정 };
viewer.openFile('sample.mrd', '/r/n [sample.txt]', options);

```

속성

옵션에는 다음의 속성을 설정할 수 있습니다. [노트](#)

[관련 속성](#)

[배율 관련 속성](#)

[UI 관련 속성](#)

[제출 관련 속성](#)

[서버 사이트 관련 속성](#)

[기타 속성](#)

노트 관련 속성

enableNote : **boolean**

노트 기능의 활성화 여부

설정 값 : true, false(기본값)

includeNoteInPrint : boolean

인쇄물에 노트 포함 여부

설정 값 : true, false, undefined(기본값)

includeNoteInExport : boolean

저장 파일에 노트 포함 여부

설정 값 : true, false, undefined(기본값)

excludeNoteTypeInPrint : Array

인쇄물에 노트 포함할 때 제외할 옵션 설정

설정 값 : ['text' (텍스트 노트) | 'scribble' (펜과 형광펜)]

```
// 인쇄물에 텍스트 노트와 펜,형광펜을 포함하지 않음
{ enableNote : true, excludeNoteTypeInPrint : [ 'text', 'scribble' ] }
// 인쇄물에 텍스트 노트는 제외
{ enableNote : true, excludeNoteTypeInPrint : [ 'text' ] }
// 인쇄물에 펜과 형광펜은 제외
{ enableNote : true, excludeNoteTypeInPrint : [ 'scribble' ] }
```

excludeNoteTypeInExport : Array

저장 파일에 노트 포함 여부

설정 값 : ['text' (텍스트 노트) | 'scribble' (펜과 형광펜)]

```
// 저장파일에 노트와 펜,형광펜을 포함하지 않음
{ enableNote : true, excludeNoteTypeInExport : [ 'text', 'scribble' ] }
// 저장파일에 텍스트 노트는 제외
{ enableNote : true, excludeNoteTypeInExport : [ 'text' ] }
// 저장파일에 펜과 형광펜은 제외
{ enableNote : true, excludeNoteTypeInExport : [ 'scribble' ] }
```

scrollbarOnNote : object

노트 모드에서 스크롤바의 두께와 색상을 지정

설정 값 : { width, color }

width : number

스크롤바 두께 지정

기본 값 : 18px

color : string

스크롤바 색상 지정

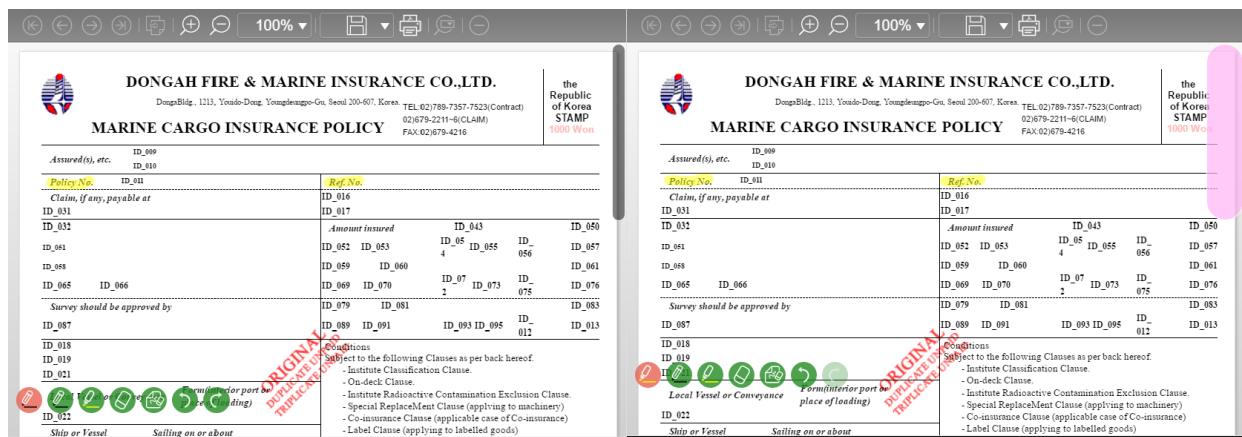
기본 값 : rgba(0, 0, 0, 0.498039)

관련 속성 : [뷰어 스크롤 속성 -scrollbarOnViewer](#)

예

노트 모드로 수행하며 노트 모드에서의 스크롤바 두께와 컬러를 지정할 경우 아래와 같이 사용합니다.

```
<script>
window.onload = function(){
    var viewer = new
    m2soft.crownix.Viewer( 'http://localhost:8080/Re
    portingServer/service', 'crownix-viewer');
    viewer.openFile(
        'sample.mrd', '/rfrn [sample.txt]',
        {enableNote : true, scrollbarOnNote: { width: '50px', color: 'rgba(255,178,245,0.7)}});
    };
</script>
```



<아무 옵션을 주지 않은 경우 (default)>

<50px, pinkcolor의 옵션을 적용한 경우>

배율 관련 속성

defaultZoom : number | string

초기 배율 지정

설정 값 :

0.25~2.5 사이의 값

undefined(기본 값)

'width', 'height' : 가로 맞춤과 세로 맞춤 (이 경우에는 따옴표를 포함하여 지정)

defaultZoomCentre : string

초기 배율 지정 시 초기 위치 지정하는 옵션으로 defaultZoom을 함께 사용해야 함 설정

값 :

'LEFTTOP' : 보고서의 상단 왼쪽 부분을 기준으로 표시

'CENTER' : 보고서의 중앙 부분을 기준으로 표시(기본값)

keepZoomRatio : boolean

사용자가 설정한 배율을 페이지 이동시에도 유지

확대, 축소 시 보고서 상단 왼쪽을 기준으로 표시

설정 값 : true, false(기본값)

예

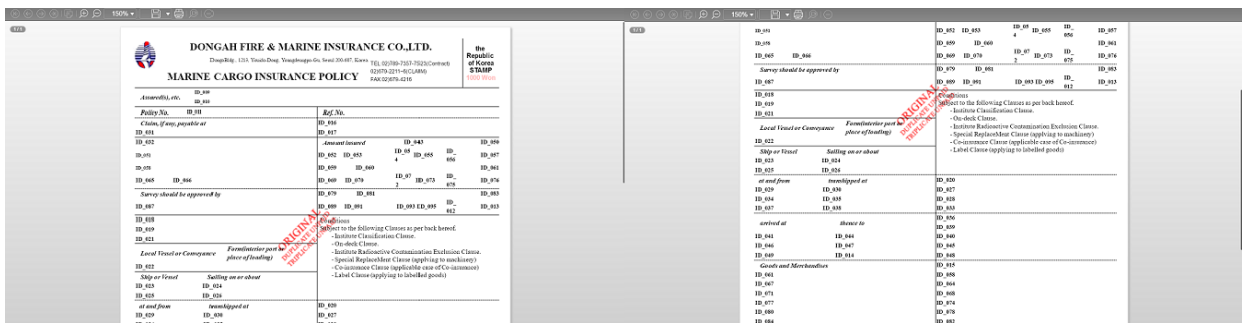
ü defaultZoom과 defaultZoomCentre 옵션

defaultZoom은 보고서의 기본 배율을 설정하고 싶을 때 사용하는 옵션입니다.

```
viewer.openFile('sample.mrd', '/rfn [sample.txt]', {defaultZoom:1.5});
```

보고서의 기본 배율을 150%으로 지정하고,상단왼쪽 영역을 기준으로 확대된 화면을 보고 싶을때, openFile 메서드 옵션 부분을 다음과 같이 변경해 줍니다.

```
viewer.openFile('sample.mrd', '/rfn [sample.txt]', {defaultZoom:1.5, defaultZoomCentre:'LEFTTOP'});
```



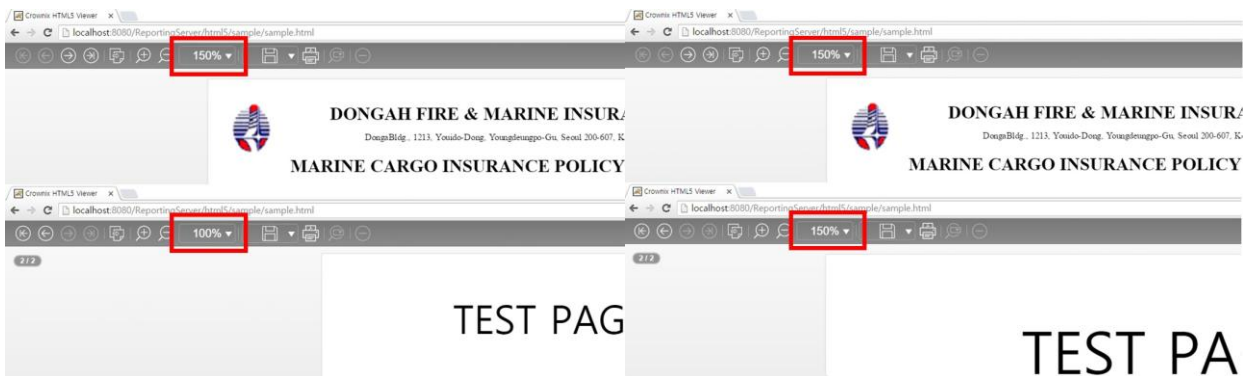
< LEFT TOP을 사용한 경우>

< CENTER을 사용한 경우(default)>

ü keepZoomRatio 옵션 (defaultZoom 및 defaultZoomCentre 옵션과 함께 사용할 수 없음)

사용자가 설정한 배율을 페이지 이동시에도 유지하고 싶을 때 openFile 메서드 옵션 부분을 다음과 같이 변경합니다.

```
viewer.openFile('sample.mrd', '/rfn [sample.txt]', {keepZoomRatio:true});
```



< false인 경우 (default) >

< true인 경우>

UI 관련 속성

팝업창 관련 UI 속성

noDataDialog : **object**

데이터가 없을 때 나타나는 대화상자의 사용자 지정

설정 값 : { show, message, title }

show : **boolean**

대화상자의 출력 여부

설정 값 : true(기본값), false

message : **string**

대화상자의 내용 (기본값은 undefined)

title : **string**

대화상자의 제목 (기본값은 undefined)

showDrilldownMsg : **boolean**

드릴다운 시 우측 로딩창 출력 여부

설정 값 : true(기본값), false

showPageNum : **boolean**

보고서가 로드된 후 좌측 상단에 나타나는 페이지 수의 출력 여부

설정 값 :

0 - 페이지 번호가 보이지 않음

1 - 3초 후 페이지 번호가 사라짐 (기본값)

2 - 페이지 번호 유지

showPageLoadingOverlay : **boolean**

페이지를 로딩할 때 로딩 이미지 출력 여부

설정 값 : true, false (기본값)

scrollbarOnViewer : object

보고서의 스크롤바의 두께와 색상을 지정해서 사용자 정의 스크롤로 보여주는 옵션입니다.

설정 값 : { width, color }

width : *number*

스크롤바 두께 지정

기본 값 : 7px

color : *string*

스크롤바 색상 지정

기본 값 : rgba(0, 0, 0, 0.498039) rgba: 투명도 a가 추가된 rgb

관련 속성 : [노트 관련 스크롤 속성 -scrollbarOnNote](#)

scrollOptions : object

스크롤바의 터치 및 마우스 이벤트를 제어하는 기능

설정 값 : { disableTouch, disablePointer, disableMouse, zoomMin, zoomMax, doubleTap , pageMargin, useInfiniteScroll}

disableTouch : *boolean*

터치 이벤트 허용 여부.

true 로 설정하면 터치 이벤트가 동작하지 않음.

설정 값 : true, false(기본값)

disablePointer : *boolean*

포인터 이벤트 허용 여부.

true 로 설정하면 포인터 이벤트가 동작하지 않음.

설정 값 : true, false(기본값)

disableMouse : *boolean*

마우스 이벤트 허용 여부.

true 로 설정하면 마우스 이벤트가 동작하지 않음.

설정 값 : true, false(기본값)

zoomMin : *number*

문서의 최소 축소 비율

설정 값 : 0.1 ~ 10 (기본값 : 0.25)

zoomMax : *number*

문서의 최대 확대 비율

설정 값 : 0.1 ~ 10 (기본값 : 2.5)

doubleTap : *boolean*

모바일에서 더블터치로 확대 /축소 사용 여부

설정 값 : true (기본값), false

pageMargin : *number*

페이지 여백 설정 -툴바 아래 페이지 표시 영역 에서의 백.

0을 설정하면 투바에 딱 붙는 것으로 보일 수 있습니다.

세로 스크롤을 사용할 때 페이지 간격으로 설정될 수 있습니다.

기본 값 : 12 (0보다 작은 값은 0으로 설정됨)

useInfiniteScroll : *boolean*

세로 스크롤 사용 여부

설정 값 : true, false (기본값)

참고사항 : 자세한 내용은 보고서 작성 가이드 > 성능 관련 옵션 > [보고서 스크롤 이벤트](#) 참조

제약사항 : 확대 관련 기능 사용시 투바의 메뉴는 변경 선택되지 않습니다.

scrollMovePage : *boolean*

마우스 휠과 PageDown, PageUp 키로 페이지 이동 여부 지정

설정 값 : true(기본값), false

useScrollTransition : *boolean*

부드러운 스크롤링 사용 여부

설정 값 : true(기본값), false

swipe : *object*

페이지 스와이프에 대한 사용자 지정

설정 값 : { useOnZoom, sensitivity, useOnDesktop }

useOnZoom : *boolean*

확대가 되어있는 상태에서 페이지 스와이프 사용 여부

설정 값 : true(기본값), false

sensitivity : *number*

페이지 스와이프 민감도 지정

설정 값 : 1~3 사이의 값

useOnDesktop : *boolean*

데스크탑환경에서 모바일환경의 스와이프 기능 사용 여부

설정 값 : true, false(기본값)

useTableOfContents : *boolean* | *object*

뷰어 화면에 슬라이드 메뉴가 나타나며 목차가 표시됩니다.

보고서에 목차를 지정하고 사용할 수 있습니다.

설정 값 : true, false(기본값), { *highlight* , *merge* }

highlight : *object*

설정 값 : { *tree*, *field* }

tree : *string*

슬라이드 탭 목차 트리 중 선택 필드의 표시 색상을 지정 (기본 값 : #00D8FF)

field : *string*

보고서내의 필드 영역 테두리 색상을 지정 (기본 값 : #AFE1FF)

pageSync : *boolean*

페이지가 이동되면 관련된 목차 노드가 하이라이트 되는 옵션.

mergeNode : *boolean*

최하위 노드 (leaf node)가 같은 이름으로 여러 개 표시될 때 하나의 노드만 표시하는 옵션입니다.

상위 노드가 같은 노드끼리 적용됩니다.

노드가 1개이고 상위 노드와 이름이 같으면 상위 노드만 표시합니다. 설정

값 : true, false (기본값)

```
viewer.openFile('sample.mrd', '/rfn [sample.txt]', { useTableOfContents : true });
viewer.openFile('sample.mrd', '/rfn [sample.txt]',
  { useTableOfContents : {
    highlight : { tree: '#FF0000', field: '#0000FF', pageSync : true }, mergeNode : true }
  }
);
```

showThumbnail : *boolean*

뷰어 화면에 슬라이드 메뉴가 나타나며 썸네일이 표시됩니다.

useTableOfContents 옵션을 함께 쓰면 목차와 썸네일이 슬라이드 메뉴에 함께 나타나며 탭으로 전환할 수 있습니다.

설정 값 : true, false(기본값)

slide : array[string]

썸네일, 목차, 유효성 검사 결과 목차로 나타나는 슬라이드 메뉴와 관련된 UI 속성입니다. 설정

값 : { closeOnSelect }

closeOnSelect : *boolean*

슬라이드 메뉴에서 한 항목을 선택하면 슬라이드가 자동으로 접히는 옵션입니다.

```
viewer.openFile('sample.mrd', '/rfn [sample.txt]', { slide: { closeOnSelect: true } });
```

disableToolBarItemsOnError : array[string]

에러 발생시 툴바 아이템 비활성화를 위한 기능

설정 값 : [툴바의 아이템 ID](#) 문자열 리스트

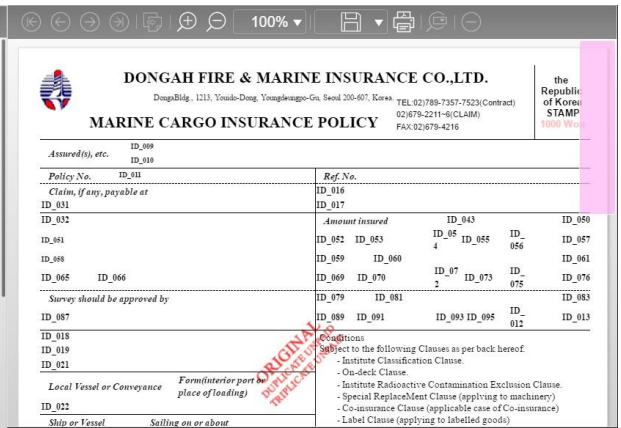
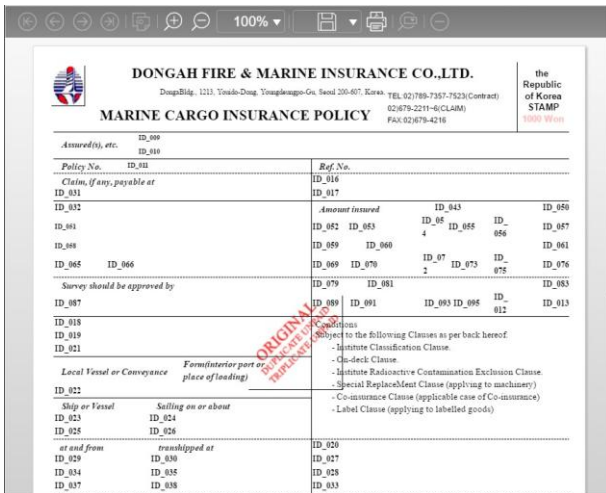
예

↳ scrollbarOnViewer 옵션

```
viewer.openFile('sample.mrd', '/rfn [sample.txt]', { scrollbarOnViewer: { width: '50px', color: 'rgba(255, 178, 245, 0.7)'} });
```

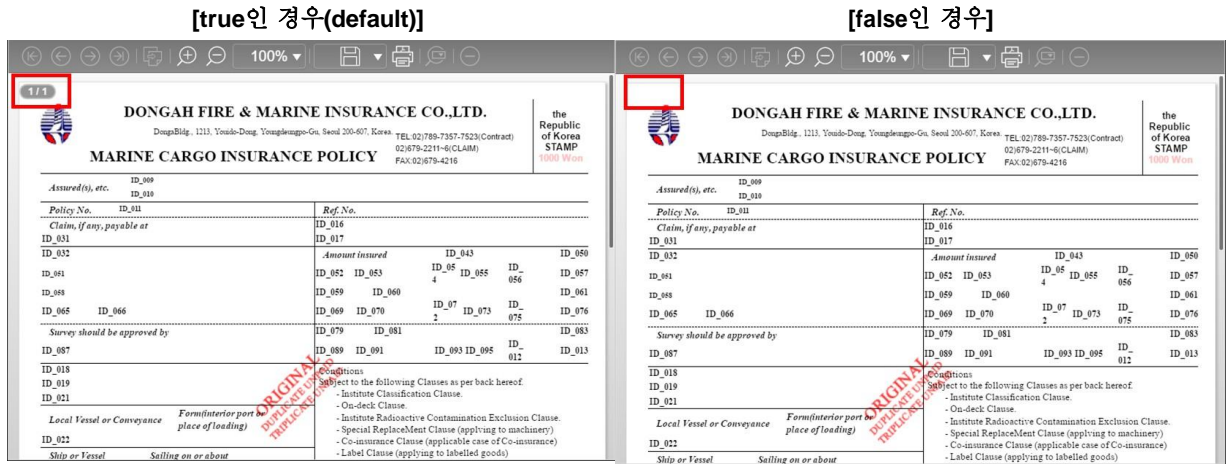
[옵션을 설정하지 않은 경우(default)]

[50px, 색상은 pink로 옵션을 적용한 경우]



↳ showPageNum 옵션

```
viewer.openFile('sample.mrd', '/rfn [sample.txt]', { showPageNum: false });
```



🔍 scrollOptions 옵션

두 핀치 줌 기능으로 축소 확대의 비율이 최소 20%, 최대 300%으로 동작합니다.

```
viewer.openFile('sample.mrd', '/rfn [sample.txt]', {zoomMin: 0.2, zoomMax: 3});
```

기타 속성

imageDrawOption : object

이미지 출력 옵션

설정 값 : { useCanvas }

useCanvas : boolean

이미지의 출력 방식(false를 사용하면 img 태그로 삽입됨)

설정 값 : true (기본값), false

layer : object

레이어 동작 옵션

설정 값 : { preload }

preload : boolean

보고서의 레이어를 언제 로드할지 선택

설정 값 :

true : 보고서의 레이어가 다른 페이지에 있더라도 사전에 로드

false (기본값) : 레이어가 있는 페이지에 이동했을 때 레이어의 내용을 로드

레이어를 로드하지 않고 스마트 폼의 제출 기능을 사용하면 레이어에 포함된 폼필드는 제출될 수 없습니다.
페이지를 모두 보지 않고 레이어의 기본값까지 포함해서 제출하려면 preload를 true로 설정하십시오.

pdfReaderNotFoundMessage : string

IE에서 adobe viewer를 찾지 못했을 경우에 보여줄 메시지를 지정

clientSidePdf : object

클라이언트 사이드로 PDF 생성하기 위한 옵션

설정 값 : { downloadFile, print, export, embeddedFont, compress }

downloadFile : boolean

PDF 저장시 클라이언트 사이드 PDF로 동작할지 여부 ([downloadFile](#) API 사용 포함)

설정 값 : true, false (기본값)

print : boolean

인쇄 실행시 클라이언트 사이드 PDF로 동작할지 여부 ([print](#) API 사용 포함)

설정 값 : true, false (기본값)

export : boolean

[export](#) API를 사용할 때 이미지 방식의 PDF로 만들지 여부 설정

값 :

true : 폰트 정보가 포함된 텍스트로 처리 (기본값)

false : 텍스트를 이미지로 처리

embeddedFont : boolean

폰트를 포함한 PDF로 저장할지 여부.

설정 값 :

true : 보고서에 사용된 폰트 수록

false : 폰트를 포함하지 않으므로 시스템에 폰트가 존재한다는 가정하에 사용 (기본값)

ü 참고사항 : PDF에 적용할 폰트는 Reporting Server에서 구성하며 서버의 환경 설정으로 폰트 캐싱을 할 수 있습니다.

compress : boolean

PDF의 내용과 폰트 정보를 압축해서 저장하기 위한 옵션입니다.

설정 값 : true (기본값), false

downloadFileName : string

저장할 파일명을 지정하는 옵션

참고사항 : [setDownloadFileName\(\)](#) 함수로도 파일명을 변경할 수 있습니다.

printDialog : object

인쇄 다이얼로그 사용을 위한 옵션

clientSidePdf : boolean

[clientSidePdf](#) 를 사용할 때 인쇄 다이얼로그 사용 여부

설정 값 : true | false (기본 값)

maxCopy : boolean

부수 설정시 최대로 설정할 수 있는 한계 값

showXlsKeepFormDialog : boolean

보고서를 Excel 파일로 저장 시 보고서 레이아웃 유지 팝업창을 띄울지 여부 지정 설정

값 : true, false(기본값)

printDialog : string

클라우드 인쇄의 스폴명을 변경시키기 위한 기능

주의사항 : ReportingServer의 매핑 정보를 변경해야 하며, 일부 브라우저에서 지원하지 않을 수 있음

useFixedFrame : **boolean**

틀고정 설정이 있는 보고서를 HTML5 Viewer에서 동작하도록 설정하는 옵션

설정 값 : true, false(기본값)

textOnCanvas : **boolean**

다량의 텍스트가 포함된 보고서의 속도 성능 개선을 위한 기능

각각의 HTML 속성을 포함한 텍스트를 단순한 Canvas 이미지로 생성할지 여부를 지정합니다.

설정 값 : true, false(기본값)

참고사항 : 자세한 내용은 보고서 작성 가이드 > 성능 관련 옵션 > [다량의 텍스트가 포함된 보고서](#) 참조

searchOption : **string**

검색 기능 중 서버사이드 검색 방식의 옵션입니다.

설정 값 :

'all' : 서버에서 페이지별로 검색어의 상세 내용을 모두 조회

'page' : 서버에서 검색어가 포함된 페이지 정보만 조회하고 상세 내용은 클라이언트에서 조회 (기본값)

참고사항 : [보고서 실행방식에 따라 다른 검색 방식](#)

fingerScan : **object**

지문 인식 기능과 관련한 옵션을 지정합니다.

설정 값 : { highQuality, documentId, serverUrl, image }

highQuality : **boolean**

지문 스캔시 고품질 사용 여부

설정 값 : true(기본값), false

documentId : **string** (필수)

문서 ID 지정

serverUrl : **string**

지문 인식 서버 주소 지정

image : **string**

지문이 정상적으로 스캔되면 나타낼 Base64로 변환된 이미지 URI

viewCache : **boolean**

캐싱 기능이 동작하도록 설정하는 옵션

뷰어로 보고서를 조회할 때 캐싱된 보고서를 조회합니다. 설정

값 : **true**, **false** (기본값)

exportCache : **boolean**

저장이나 인쇄와 관련한 기능도 캐싱된 보고서를 사용하도록 설정하는 옵션

캐싱 기능을 사용하여 조회한 보고서를 **PDF** 등의 파일로 저장할 때 동일하게 캐싱된 보고서를 사용하려면 **true**로 설정합니다.

저장하거나 인쇄할 때 캐싱된 보고서를 사용하지 않고 새로 실행시키려면 **false** 로 설정합니다.

viewCache가 **true**일 때 사용할 수 있습니다.

설정 값 : **true**, **false** (기본값)

Page

실행된 보고서의 각 페이지를 이동할 때 사용하는 API를 제공합니다.

또한, 조건에 따라 페이지를 감추거나 표시할 때 유용한 API를 제공합니다.

[nextPage](#)

[prevPage](#)

[movePage](#)

[moveFirstPage](#)

[moveLastPage](#)

[hidePage](#)

[showPage](#)

nextPage()

정의

nextPage()

설명

다음 페이지로 이동합니다.

생성되지 않은 페이지로 이동하려고 하면 경고 메시지 창이 나타납니다.

예

버튼을 추가하여 실행된 보고서의 다음 페이지로 이동하는 예시입니다.

```
<div id="nextPage">Go</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    var viewer = new
    m2soft.crownix.Viewer( 'http://localhost:8080/R
    eportingServer/service', 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    $("#nextPage").click(function(){ viewer.nextPage
    ();
    });
  };
</script>
```

prevPage()

정의

prevPage()

설명

이전 페이지로 이동합니다.

맨 처음 페이지에서 이전으로 이동하려고 하면 경고 메시지 창이 나타납니다.

예

버튼을 추가하여 실행된 보고서의 이전 페이지로 이동하는 예시입니다.

```
<div id="prevPage">Go</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
  var viewer = new
  m2soft.crownix.Viewer( 'http://localhost:8080/R
  eportingServer/service', 'crownix-viewer');
  viewer.openFile('sample.mrd', '/rfn [sample.txt]');
  $("#prevPage").click(function(){ viewer.prevPag
  e();
  });
};
</script>
```

movePage(pageNumber)

정의

movePage(number pageNumber)

설명

파라미터로 받은 페이지 번호로 이동합니다.

생성되지 않은 페이지 또는 첫 페이지 이전으로 이동하려고 하면 경고 메시지 창이 나타납니다.

인수

pageNumber : number

이동할 페이지 번호

예

버튼을 추가하여 실행된 보고서의 특정 페이지(2번째 페이지)로 이동하는 예시입니다.

```
<div id="movePage">Go</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
    var viewer = new
    m2soft.crownix.Viewer( 'http://localhost:8080/R
    eportingServer/service', 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    $("#movePage").click(function(){ viewer.movePag
    e(2);
    });
};
</script>
```

moveFirstPage()

정의

moveFirstPage()

설명

첫 페이지로 이동합니다.

아직 페이지가 생성되지 않았을 때 이동하려고 하면 경고 메시지 창이 나타납니다.

예

버튼을 추가하여 실행된 보고서의 첫 페이지로 이동하는 예시입니다.

```
<div id="moveFirstPage">Go</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
    var viewer = new
    m2soft.crownix.Viewer( 'http://localhost:8080/R
    eportingServer/service', 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    $("#moveFirstPage").click(function(){ viewer.move
    FirstPage();
    });
};
</script>
```

moveLastPage()

정의

moveLastPage()

설명

마지막 페이지로 이동합니다.

아직 페이지가 생성되지 않았을 때 이동하려고 하면 경고 메시지 창이 나타납니다.

예

버튼을 추가하여 실행된 보고서의 마지막 페이지로 이동하는 예시입니다.

```
<div id="moveLastPage">Go</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
    var viewer = new
    m2soft.crownix.Viewer( 'http://localhost:8080/R
    eportingServer/service', 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    $("#moveLastPage").click(function(){ viewer.move
    LastPage();
    });
};
</script>
```

moveFormField(FormFieldID)

정의

moveFormField(String formfield id)

설명

파라미터로 받은 폼 ID의 폼필드가 있는 페이지로 이동 후 해당 폼필드에 포커스를 줍니다.

존재하지 않은 폼 ID의 폼필드로 이동하려고 하면 경고 메시지 창이 나타납니다.

인수

formFieldId : string

이동하려는 폼ID 및 필드 ID 정보

"폼ID.폼필드ID"의 형식으로 기재

예

보고서 로딩이 완료되면 보고서의 특정 폼필드로 이동하는 예시입니다.

```
<div id="moveFormField">Go</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
    var viewer = new
    m2soft.crownix.Viewer( 'http://localhost:8080/R
    eportingServer/service', 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    $("#moveFormField").click(function(){ viewer.mov
    eFormField("form1.check1");
    });
};
</script>
```

hidePage(pageNumber)

정의

hidePage(number pageNumber)

설명

지정된 페이지를 화면에 표시하지 않습니다.

인수

pageNumber : **number**

감출 페이지 번호를 1개 이상 지정합니다.

예

버튼을 추가하여 실행된 보고서의 다음 페이지를 표시하지 않는 샘플입니다.

```
<div id="hidePage">Hide</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    var viewer = new
    m2soft.crownix.Viewer( 'http://localhost:8080/R
    eportingServer/service', 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    $("#hidePage").click(function(){ viewer.hidePage(
    2,3,4);
    });
  };
</script>
```

showPage(pageNumber)

정의

showPage(number pageNumber)

설명

[hidePage\(pageNumber\)](#) 로 감춘 페이지를 다시 표시합니다.

인수

pageNumber : number

보여줄 페이지 번호를 1개 이상 지정합니다.

예

버튼을 추가하여 실행된 보고서의 일부 페이지를 표시하는 샘플입니다.

```
<div id="showPage">Show</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
    var viewer = new
    m2soft.crownix.Viewer( 'http://localhost:8080/R
    eportingServer/service', 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    $("#showPage").click(function(){ viewer.showPag
    e(2,3,4);
    });
};
</script>
```


Zoom

실행된 보고서의 확대/축소와 관련된 API를 제공합니다.

[zoomIn](#)

[zoomOut](#)

[pack\(\)](#)

zoomIn()

정의

zoomIn()

설명

페이지를 현재 비율보다 20% 확대합니다.

최대 250% 까지 확대됩니다.

반환 값

ratio : number

보고서의 배율 0.25~2.5 내의 값을 반환 (1 : 100% 를 의미)

예

버튼을 추가하여 ,실행된 보고서를 확대하여 보기 위한 예시입니다.

```
<div id="zoomIn">Zoom In</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
    var viewer = new
    m2soft.crownix.Viewer( 'http://localhost:8080/R
    eportingServer/service', 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    $("#zoomIn").click(function(){
        viewer.zoomIn();
    });
};
</script>
```

zoomOut([ratio])

정의

zoom()

zoom(ratio)

설명

페이지를 비율에 맞게 확대 또는 축소합니다.

0.25 ~ 2.5 비율의 범위 내에서 확대 축소합니다.

인수

ratio : number | string

확대 또는 축소할 비율

설정 값 :

0.25~2.5 내의 값 (1 : 100% 를 의미)

'height' | 'width' (세로맞춤 또는 가로맞춤)

반환 값

ratio : number

보고서의 배율 0.25~2.5 내의 값을 반환 (1 : 100% 를 의미)

예

버튼을 추가하여, 실행된 보고서를 축소하여 보기 위한 예시입니다.

```

<div id="zoomOut">Zoom Out</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
  var viewer = new
  m2soft.crownix.Viewer( 'http://localhost:8080/Re
  portingServer/service', 'crownix-viewer');
  viewer.openFile('sample.mrd', '/rfn [sample.txt]');
  $("#zoomOut").click(function(){ viewer.zoomOu
  t();
  });
};
</script>

```

버튼을 추가하여, 실행된 보고서를 50%로 축소하여 보기 위한 예시입니다.

```
<divid="zoom">Zoom</div>
<divid="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    var viewer = new
      m2soft.crownix.Viewer( 'http://localhost:8080/Re
        portingServer/service', 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    $("#zoom").click(function(){
      viewer.zoom(0.5);
    });
  };
</script>
```

pack()

분류

viewer

정의

pack()

설명

보고서 페이지의 확대/축소 비율을 가로 맞춤으로 확대하거나 축소하기 위한 API 입니다.

예

버튼을 추가하여, 확대/축소된 보고서의 페이지를 가로 맞춤을 하는 예제 입니다.

```
<div id="pack">pack</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
    var viewer = new
    m2soft.crownix.Viewer( 'http://localhost:8080/R
    eportingServer/service', 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    $("#pack").click(function(){ vi
    ewer.pack();
    });
};
</script>
```

Toolbar

HTML5 Viewer 실행시 나타나는 툴바와 포함되는 툴바 아이템의 표시 여부를 결정하기 위한 API를 제공합니다.

[hideToolbar](#)

[showToolbar](#)

[hideToolbarItem](#)

[showToolbarItem](#)


ToolbarItem

HTML5 Viewer의 툴바에는 하위 아이템이 있으며, 각 아이템은 ID와 아이콘이 있습니다. 이 ID를 이용하여 툴바에서 일부 메뉴를 숨기거나 보이게 하는 API를 제공합니다.

m 툴바의 아이템 ID

각 아이템은 ID를 가지며, 관련 API에서 사용됩니다.

ID의 타입은 string이며, 아이콘 명과 매칭되는 아이콘은 아래와 같습니다.

ID	아이콘	이름	기능
first		첫 페이지	첫 페이지로 이동합니다. 현재 페이지가 첫 페이지일 경우, 첫 페이지 버튼이 비활성화 됩니다.
prev		이전 페이지	이전 페이지로 이동합니다. 현재 페이지가 첫 페이지일 경우, 이전 페이지 버튼이 비활성화 됩니다.
next		다음 페이지	다음 페이지로 이동합니다. 현재 페이지가 마지막 페이지일 경우, 다음 페이지 버튼이 비활성화 됩니다.
last		마지막 페이지	마지막 페이지로 이동합니다. 현재 페이지가 마지막 페이지일 경우, 마지막 페이지 버튼이 비활성화 됩니다.
move		페이지 이동	페이지 이동 버튼을 누르면 페이지 이동 창이 뜹니다. 원하는 페이지 번호를 입력한 후 확인 버튼을 누르면 해당 페이지로 이동합니다.
zoomin		확대	보고서를 현재의 배율에서 확대하여 보여줍니다. 누를 때 마다 20%씩 확대됩니다.
zoomout		축소	보고서를 현재의 배율에서 축소하여 보여줍니다. 누를 때 마다 20%씩 축소됩니다.
ratio		배율	보고서를 선택한 배율로 확대/축소합니다. (하위 선택 배율 제공)
25	25%	배율(25%)	25% 배율로 확대/축소합니다.
50	50%	배율(50%)	50% 배율로 확대/축소합니다.
75	75%	배율(75%)	75% 배율로 확대/축소합니다.
100	100%	배율(100%)	100% 배율로 확대/축소합니다.
125	125%	배율(125%)	125% 배율로 확대/축소합니다.
150	150%	배율(150%)	150% 배율로 확대/축소합니다.
200	200%	배율(200%)	200% 배율로 확대/축소합니다.
250	250%	배율(250%)	250% 배율로 확대/축소합니다.
width	가로 맞춤	가로 맞춤	가로 맞춤으로 확대/축소합니다.
height	세로 맞춤	세로 맞춤	세로 맞춤으로 확대/축소합니다.

save		저장	보고서를 PDF, DOC, XLS, PPT, HWP의 포맷으로 저장할 수 있습니다. 각 포맷의 저장 버튼을 누르면 저장 메시지 출력 후 다운로드 화면이 나옵니다. 다운로드 받는 화면은 브라우저마다 다를 수 있습니다.
pdf		저장(PDF)	PDF 형식의 파일로 저장합니다.
doc		저장(DOC)	DOC 형식의 파일로 저장합니다.
xls		저장(XLS)	XLS 형식의 파일로 저장합니다.
ppt		저장(PPT)	PPT 형식의 파일로 저장합니다.
hwp		저장(HWP)	HWP 형식의 파일로 저장합니다.
storage		보고서	한 번 조회했던 보고서 서식(MML)이나 작성 중이던 폼 데이터를 브라우저에 임시 저장했다가 다시 불러올 수 있습니다.
print		브라우저 인쇄	브라우저의 인쇄 기능을 이용하여 화면에 표시된 보고서를 인쇄합니다. 브라우저마다 인쇄 품질이 다를 수 있습니다.
cloud_print		클라우드 인쇄	구글 클라우드 인쇄 기능을 이용하여 보고서를 인쇄합니다. 클라우드 프린트가 설정이 되어 있어야 합니다.
print_pdf		PDF 인쇄	보고서를 서버에서 PDF로 변환 후 PDF 뷰어를 이용하여 보고서를 인쇄합니다. 브라우저에 상관없이 인쇄 품질이 동일하지만 브라우저 설정에 따라 PDF 미리 보기가 표시되지 않을 수 있습니다.
inquiry		조회 툴바	조회 툴바 기능을 사용할 수 있습니다. 이 기능은 보고서를 만들 때 Designer에서 설정된 조회 툴바 조건에 따라 사용할 수 있습니다.
cancel		리포팅 중지	HTML5 Viewer는 서버에서 보고서를 만들면서 사용자에게 보여줍니다. 따라서 보고서가 생성 중이더라도 화면에 첫 번째 장부터 보여주기 시작합니다. 만약 원하는 곳까지 보고서 생성이 되었거나 더 이상 볼 필요가 없어졌다면 리포팅 중지 버튼을 눌러 서버에서 다운로드 받는 작업을 멈출 수 있습니다.
close		overlay 모드에서 뷰어 닫기	기본적으로 overlay 모드에서는 팝업된 창 오른쪽 끝에 빨간색 동그라미의 x 버튼으로 닫습니다. 이 툴바 아이템을 활성화 하면 x 버튼이 사라지고 툴바의 제일 앞에 닫기 아이콘이 나타나며 x 버튼과 같은 기능을 수행합니다.
sharing		폼 셰어링 사용	셰어링 기능을 사용 할 수 있습니다. 보고서를 공유 할 수 있게해주는 기능을 수행합니다.
lock		보고서 셰어링 잠금	방을 개설한 호스트가 보고서를 작성 중 일때,클라이언트나 허용되지 않은 사용자가 보고서를 수정 하지 못 하도록 잠금하는 기능입니다.

hideToolbarItem(items)

정의

hideToolbarItem(string[] items)

설명

툴바의 아이템 중 숨기고자 하는 아이템을 지정합니다.

인수

items : array (string[])

[툴바의 아이템 ID](#) 의 배열

설정 값 : [itemID, itemID, ...]

예

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.hideToolbarItem(["ratio", "save"]);
  };
</script>
```

showToolbarItem(items)

정의

showToolbarItem(array items)

설명

툴바의 아이템 중 표시하고 하는 아이템을 지정합니다.

인수

items : array (string[])

[툴바의 아이템 ID](#) 의 배열

설정 값 : [itemID, itemID, ...]

예

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.showToolbarItem(["ratio", "save"]);
  };
</script>
```

addToolbarItem(item)

정의

addToolbarItem(object item)

설명

툴바에 사용자 정의 아이템을 추가합니다.

인수

item : object

사용자 정의 투바 속성

설정 값 : { id, index, svg, title, separator, handler }

id : string

툴바 아이템의 고유 ID

showToolbarItem, hideToolbarItem 을 사용할 때 식별자로 이용

중복아이디를 사용할 경우 예외가 발생

index : number

현재 투바에서 index 번째에 아이템을 삽입

0보다 작을 경우 : 첫번째에 아이템을 삽입

잘못된 숫자 또는 현재의 아이콘 개수보다 큰 숫자 : 마지막에 아이템을 삽입

svg : string

툴바에 사용할 SVG 아이콘

크기 : 가로 -32px, 세로 -32px

title : string

툴바 아이콘의 투팁으로 사용될 문자열

separator : boolean

아이콘의 오른쪽에 구분자를 삽입하는 것의 여부

handler : function

툴바를 클릭했을 경우 실행될 함수

예

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
var myItem =
{ index : 1,
  id : 'myItem',
  svg : '<svg version="1.1" xmlns="http://www.w3.org/2000/svg'
```

```

xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
width="32px" height="32px"></svg>',
title : '나의 아이템',
separator : true,
handler : function() {
    alert('myItem click!');
}
};
window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.showToolBarItem(["ratio", "save"]);
    viewer.addItem(myItem);
};
</script>

```

hideNoteItem(items)

정의

hideNoteItem(String[] items)

설명

노트의 아이템 중 숨기고자 하는 아이템을 지정합니다.

인수

items : array (string[])

노트 아이템의 배열

설정 값 : ['pen', 'highlighter', 'textNote', 'eraser', 'eraseAll', 'undo', 'redo']

주의 사항

사용하기 전, 노트 기능을 활성화해야 합니다.

모든 아이템을 숨길 경우, 펜이 기본값으로 동작합니다.

예

```

<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.openFile('sample.mrd', '/r/n [sample.txt]', {enableNote : true});
    viewer.hideNoteItem(['pen', 'textnote', 'eraserAll']); // 펜, 텍스트 노트, 페이지 노트 삭제 기능을 숨깁니다.
};
</script>

```

showNoteltem(items)

정의

showNoteltem(String[] items)

설명

노트 아이템 중 표시하고 하는 아이템을 지정합니다.

인수

items : array (string[])

노트 아이템의 배열

설정 값 : ['pen', 'highlighter', 'textNote', 'eraser', 'eraseAll', 'undo', 'redo']

주의 사항

사용하기 전, 노트 기능을 활성화해야 합니다.

예

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
  var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
  viewer.openFile('sample.mrd', '/rfn [sample.txt]', {enableNote : true});
  viewer.hideNoteltem(['pen', 'highlighter', 'textnote', 'eraser', 'eraserAll', 'undo', 'redo']); // 전체 기능을 숨깁니다.
  viewer.showNoteltem(['pen', 'textnote', 'eraserAll']); // 펜, 텍스트 노트, 페이지 노트 삭제 기능을 보여줍니다.
};
</script>
```

hideToolbar()

정의

hideToolbar()

설명

툴바를 숨깁니다.

예

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.hideToolbar();
  };
</script>
```

showToolbar()

정의

showToolbar()

설명

툴바를 표시합니다.

예

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.showToolbar();
  };
</script>
```

dockToolbar(direction)

정의

```
dockToolbar(string direction)
```

설명

툴바를 한 쪽 방향으로 도킹합니다.

기본적으로 'top'으로 도킹되어 있으며 그 외의 다른 방향으로도 위치시킬 수 있습니다.

인수

direction : string

툴바가 위치할 방향

설정 값 : 'left'(기본값), 'top', 'right', 'bottom'

예

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.dockToolbar('left');
  };
</script>
```

floatToolbar()

정의

floatToolbar()

설명

보고서를 실행했을 때 툴바가 나타나지 않고, 보고서 영역을 클릭하지 하면 툴바가 나타납니다. 툴바가 나타난 상태에서 다시 클릭하지 하면 툴바가 사라집니다.

예

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.floatToolbar();
  };
</script>
```


stickToolbar()

정의

stickToolbar()

설명

툴바가 항상 특정 영역을 차지하여 나타납니다.

반대 되는 개념인 [floatToolbar\(\)](#)로 투바를 항상 표시하지 않도록 했을 때, 다시 되돌릴 수 있습니다.

예

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.stickToolbar();
  };
</script>
```

Export & Import

실행된 보고서를 여러가지 형식으로 Export 하기 위한 API를 제공합니다.

Export 된 내용이 폼 서식인 MML이나 폼 데이터인 경우, 다시 Import 하기 위한 API도 제공합니다.

[export\(type, callback\)](#)

[exportMML\(options\)](#)

[exportFormData\(options\)](#)

[importMML\(options\)](#)

[importFormData\(options\)](#)

주의 사항

이 기능은 [Storage 기능이 포함된 viewer 모듈](#)을 사용해야 정상 동작합니다.

externalStorage(option, callback, openFileOpt)

정의

externalStorage(option, callback, openFileOpt)

설명

실행된 보고서 또는 폼 데이터를 옵션에 따라 서버사이드로 저장하거나 조회합니다.

인수

options : object

동작 방식을 결정하고 포함할 데이터를 결정합니다.

설정 값 : {method, mml, formData} | {method, mml, formData, key, timeout }

method : string

동작 방식 (필수값)

설정 값 :

'PUT' : 서버사이드로 저장합니다.

'GET' : 서버의 데이터를 가져와서 보고서에 반영합니다.

'DELETE' : 서버의 데이터를 삭제합니다.

'EXIST' : 서버에 데이터가 있는지 확인합니다.

mml : boolean

mml 포함 여부

설정 값 : true (기본값), false

info : boolean

method가 'PUT' 일 때 부가 정보를 포함할지 여부

mml : true 로 설정된 경우에는 info에 설정된 값에 상관 없이 info : true 로 적용됩니다.

부가 정보는 [MML Object](#) 에 명시된 속성 중에서 mml 속성을 제외한 나머지 속성입니다.

설정 값 : true (기본값), false

formData : boolean

폼필드 데이터 포함 여부

설정 값 : true (기본값), false

key : string

서버에 저장될 데이터의 고유 키

key를 설정 하지 않으면 openFile의 옵션에 설정된 externalStorage의 Key 값이 기본 값이 됩니다.

timeout : *number*

timeout 시간 지정 (초 단위)

설정 하지 않으면 `openFile`의 옵션에 설정된 `infotalk`의 `timeout`의 값이 기본 값이 됩니다 (기본값 : 30)

includeNote : *boolean*

노트 포함하여 저장할지 여부

설정 값 : `true`, `false` (기본값)

callback : *function*

서버사이드 동작이 끝나고 실행되는 함수

반환 값 : 동작 결과

param : *object* { *boolean* status, *string* message, *object* isExist }

status : 제출 성공 여부 (`true` : 성공 `false` : 실패)

message : 수행 결과가 실패인 경우의 에러 메시지

isExist : { *boolean* mml, *boolean* formData }

mml : mml 포함 여부

formData : formData 포함 여부

openFileOpt : *object*

`options`의 `method`가 `GET`일 때 보고서를 실행하는 경우 사용합니다.

`openFile` API의 [options](#)와 동일합니다.

예

☹ 서버에 저장된 `mml`을 가져와서 실행하는 예시

```
var viewer = new m2soft.crownix.Viewer(
    'http://localhost:8080/ReportingServer/service',
    'crownix-viewer', { infotalkUrl : 'http://localhost:8080/InfoTalkServer' });
var doc_1_key = '1';
var openFileOptions = {
    // 속성 설정
};
viewer.externalStorage (
    { method : 'GET', mml : true, formData : true, key : doc_1_key },
    function (param) {
        console.log(param.status);
        console.log(param.message);
    },
);
```

```
openFileOptions  
);
```

☺ 서버에 폼데이터를 저장하는 예시

```
var viewer = new m2soft.crownix.Viewer(  
    'http://localhost:8080/ReportingServer/service',  
    'crownix-viewer', { infotalkUrl : 'http://localhost:8080/InfoTalkServer' });  
var openFileOptions = {  
    // 속성 설정  
    externalStorage : { key : '1' }  
};  
viewer.openFile( mrd, mrd_param, openFileOptions );  
viewer.externalStorage (   
    { method : 'PUT', mml : false, formData : true, info : true }  
);
```

export(type, callback)

정의

export(string type, function callback)

설명

실행된 보고서를 다른 파일 포맷으로 변환합니다. 일반

PDF로 저장하는 경우는 파일로 저장합니다.

이미지 형식의 PDF나 png, jpeg와 같은 일반 이미지 형식으로 저장하는 경우 Base64 데이터 형식으로 변환되며 Callback function 파라미터를 통해 리턴됩니다.

인수

type : string

저장할 파일의 타입

설정 값 :

'png' : PNG 이미지 형식

'jpeg' : JPEG 이미지 형식

'pdf' : PDF 파일 형식

callback : function

저장이 끝나고 실행되는 함수

ü 저장할 타입이 이미지인 경우

Base64로 인코딩된 배열이 파라미터로 전달됨

ü 저장타입이 PDF일 경우

Base64 로 인코딩된 PDF 문서가 전달됨

예

다음은 저장 버튼을 클릭했을 때 보고서를 pdf/Jpeg 파일로 저장하는 스크립트 예시입니다.

```

<div>
  <input type="button" value="pdf export" id="pdf" />
  <input type="button" value="jpeg export" id="jpeg" />
  <span id="result"></span>
</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  var viewer;

```

```
$(document).ready(function(){
  $('#pdf').bind('click',
    function(){ viewer.export('pdf',
      function(pdf){
        // pdf 변수에 PDF의 Base64 인코딩 결과가 반영됨
        console.log(pdf);
      });
    });
  $('#jpeg').bind('click',
    function(){ viewer.export('jpeg',
      function(imageList){
        // Export한 이미지 배열(imageList)을 새로운 창에 그려줌
        var imageView = window.open("", 'image view',
          'toolbar=no,scrollbars=yes,resizable=yes');
        var html = [];
        html.push('<html><head><title>image view</title></head><body>');
        for(var i = 0; i < imageList.length; i ++){
          html.push('<a href=javascript:window.close()><img src=');
          html.push(imageList[i]);
          html.push(' border=0 alt=close></a>');
        }
        html.push('</body></html>');

        imageView.document.open();
        imageView.document.write(html.join(""));
        imageView.document.close();
      });
    });

  viewer = new
  m2soft.crownix.Viewer( 'http://localhost:8080/R
  eportingServer/service', 'crownix-viewer');
  viewer.openFile('sample.mrd', '/r/n [sample.txt]');
});
```


exportFormData(options)

정의

string exportFormData(object options)

object exportFormData(object options)

설명

폼 편집 모드로 실행된 보고서에서 입력 중의 폼 데이터를 저장합니다. 저장 결과를 문자열 형태로 리턴합니다.

인수

options : object

저장할 때 적용할 옵션을 제공

설정값 : { checkValid, includeExtra, includeNote }

checkValid : boolean

저장 전에 유효성 검사 여부

설정값 : true, false (기본 값)

includeExtra : boolean

조회된 보고서가 이하의 정보를 포함하여 저장할지 여부

1) 보고서의 상태가 달라지는 경우의 정보

- 드릴 다운 기능 사용
- 폼필드 스크립트의 hidePage() 로 페이지 삭제

2) 현재 페이지 정보

설정값 :

true : 정보를 포함하여 저장

false : 포함하지 않음(기본값)

includeNote : boolean

조회된 보고서가 노트를 포함하여 저장할지 여부

설정값 : true, false (기본 값)

반환 값

formData : string

XML 형태로 구성된 폼필드 데이터 문자열

options 인수의 includeExtra에 false를 적용한 경우, 런타임에 변경된 보고서의 상태를 포함하지 않고 데이터만 반환

formData : object

이하의 정보를 포함한 데이터

• XML 형태로 구성된 폼필드 데이터

• 런타임에 변경된 보고서 상태 정보 -options 인수의 includeExtra에 true 적용한 경우, 런타임에 변경된 보고서의 상태를 포함하여 반환

• 노트 데이터 -options 인수의 includeNote에 true 적용한 경우, 노트를 포함하여 반환

주의 사항

[보고서 저장 기능이 포함된 js 파일](#)을 사용해야 이 API를 사용할 수 있습니다.

예

버튼을 추가하여, 실행된 보고서의 폼필드 데이터를 Export하는 예시입니다.

```
<div="export-data">Export</div>
<div="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  var viewer;
  $(document).ready(function(){
    var viewer = new
      m2soft.crownix.Viewer( 'http://localhost:8080/ReportingServer/service', 'crownix-viewer');

    $('#export-data').click(function(){
      var formData = viewer.exportFormData(
        // 유효성 검사 실행
        { checkValid : true }
      );
      // 폼 데이터의 XML을 console 에 표시
      console.log(formData);
    });
  });
</script>
```

다음은 폼 데이터의 결과 예시입니다.

```
<?xml version="1.0" encoding="UTF-8"?>
<formdata> // 최상위 태그
<version>1.0</version>
<form> // 폼이 여러 개일 수 있으므로 폼 태그를 더 뉼.
<id>EyeChart</id> // 폼 아이디
  <fields> // 모든 폼필드는 fields 태그 아래에 있다.
    <field>
      <id>Doctor</id> // 폼필드 아이디
      <value><![CDATA[Kim]]></value> // 폼필드 값 모든 값은 CDATA로 감싸 있습니다.
      <type>rb</type> // 모든 타입은 mml 문서에 정의된 타입 사용
      <selectedId>Kim1</selectedId> // 라디오 버튼의 경우 선택된 필드 ID가 필요.
    </field>
    <field> // field 태그는 폼필드 개수 만큼 반복됨
      <id>txtFU</id>
      <value><![CDATA[tsetggg]]></value>
      <type>tf</type>
    </field>
  </fields>
</form>
</formdata>
```

importFormData(formData)

정의

```
importFormData(string formData)
```

```
importFormData(object formData)
```

설명

exportFormData을 통하여 작성된 폼 데이터를 열려 있는 보고서의 각 폼 필드에 적용합니다.

인수

formData : string

XML 형태로 구성된 폼필드 데이터 문자열

formData : object

XML 형태로 구성된 폼필드 데이터 및 런타임에 변경된 보고서 상태 정보

주의 사항

[보고서 저장 기능이 포함된 js 파일](#)을 사용해야 이 API를 사용할 수 있습니다.

참고 사항

[exportFormData\(options\)](#)를 "True"로 설정하고 importFormData(formData)를 사용하면 ,현재 페이지로 이동한 상태로 보고서가 실행됩니다.

예

버튼을 추가하여 실행된 보고서의 폼필드 데이터를 Export하는 예시입니다.

```

<div id="export-data">Export</div>
<div id="import-data">Import</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
var viewer;
var formData;
$(document).ready(function(){
  var viewer = new
    m2soft.crownix.Viewer( 'http://localhost:8080/R
      eportingServer/service', 'crownix-viewer');

  $('#export-data').click(function(){
    formData = viewer.exportFormData({ checkValid : true });
  });
  $('#import-data').click(function(){
    // formData가 null이 아니라는 가정 하에 적용합니다.
    viewer.importFormData(formData);
  });
}

```

```
});  
});  
</script>
```

MML Export & Import

HTML5 Viewer의 서식인 MML을 Export하거나 Import 하는 API를 제공합니다.

MML은 XML과 유사한 형태이지만, 부가 정보를 포함한 Object 형식으로 Export하거나 Import 해야 합니다.

• MML Object

MML Object는 다음의 속성을 포함합니다.

mml : *string*

보고서 서식인 mml의 저장 결과

mrd_path : *string*

보고서와 관련된 실행 mrd

mrd_param : *string*

mrd 실행시 적용되었던 파라미터

options : *object*

openFile 당시의 옵션

[openFile\(mrdPath, mrdParam, options\)](#)의 options와 참고

preventPrintExport : *boolean*

프린트 가능 여부

useAdjustLetterSpace : *boolean*

자간 조절 여부

[useAdjustLetterSpace](#)의 내용 참고

useWebFont : *boolean*

웹 폰트 사용 여부

[useWebFont](#)의 내용 참고

isAppendReport : *boolean*

AppendReport를 사용했는지 여부

version : *string*

버전

noteData : *object* { *page* : *noteBase64* }

노트 데이터

drilldownInfo : *object*

현재 drilldown 정보

layerInfo : *string*

현재 layer 정보

useAdjustCharacterScale : *boolean*

문자 크기 조절 여부

removedPage : *Array*

현재 삭제된 페이지

exportMML(options)

정의

object exportMML(*object* options)

설명

실행된 보고서를 옵션에 따라 저장합니다.

저장 결과를 *object* 형태로 리턴합니다.

인수

options : *object*

저장할 형태와 저장할 때 이미지/바코드 객체를 포함할지를 설정

저장할 형태는, 보고서 서식만 저장하거나 작성중인 폼 데이터까지 포함하여 저장하는 형태가 있습니다.

설정값 : { *includeImage*, *includeFormData* }

includeImage : *boolean*

이미지 또는 바코드의 포함 여부

설정값 : *true*, *false* (기본 값)

includeFormData : *boolean*

폼 데이터 포함 여부

설정값 : *true*, *false* (기본 값)

includeNote : *boolean*

조회된 보고서가 노트를 포함하여 저장할지 여부

반환 값

[MML Object](#)

주의 사항

[보고서 저장 기능이 포함된 js 파일](#)을 사용해야 이 API를 사용할 수 있습니다.

예

버튼을 추가하여, 실행된 보고서를 MML로 Export하는 예시입니다.

```
<div id="export-mml">Export</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  var viewer;
  $(document).ready(function(){
    var viewer = new
      m2soft.crownix.Viewer( 'http://localhost:8080/R
        eportingServer/service', 'crownix-viewer');

    $('#export-mml').click(function(){
      var formData = viewer.exportMML({
        // 이미지 포함 저장 여부(포함)
        checkValid : true,
        // 폼 데이터 포함 저장 여부(비포함)
        includeFormData : false,
      });
      // mml 서식을 console 에 표시
      console.log(mml);
    });
  });
</script>
```

importM M L(mmlobject)

정의

importMML(object options)

설명

실행된 보고서를 옵션에 따라 저장합니다.

저장 결과를 Object 형태로 리턴합니다.

인수

[MML Object](#)

주의 사항

[보고서 저장 기능이 포함된 js 파일](#)을 사용해야 이 API를 사용할 수 있습니다.

예

버튼을 추가하여 MML을 Import하는 예시입니다.

```
<div id="export-mml">Export</div>
<div id="import-mml">Import</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  var viewer;
  var mmlresult;
  $(document).ready(function(){
    var viewer = new
      m2soft.crownix.Viewer( 'http://localhost:8080/R
        eportingServer/service', 'crownix-viewer');

    $('#export-mml').click(function(){
      mmlresult = viewer.exportMML({checkValid:true, includeFormData:false});
    });
    $('#import-mml').click(function(){
      // mmlresult가 null이 아니라는 가정하에 적용합니다.
      viewer.importMML(mmlresult);
    });
  });
</script>
```

addSubmitParameters

정의

```
addSubmitParameters(object parameters)
```

설명

제출 시 사용자 파라미터를 만들어 전달하기 위한 API 입니다.

제출할 때 제출 요청의 고유 ID를 포함하여 전송할 때도 사용할 수 있습니다.

인수

parameters : **object**

사용자 정의 파라미터

설정값 : { parameters, parameterKey }

parameters : **object**

사용할 사용자 정의 파라미터

parameterKey : **parameterValue**

파라미터에 설정할 값

주의 사항

제출 요청 ID를 포함할 때를 제외하고 커스텀 핸들러를 사용할때만 적용되며, 여러번 같은 값을 여러번 사용할 수 있습니다. 단, 중첩시 마지막에 선언한 값이 설정됩니다.

[openFile](#) 또는 [externalStorage](#) (GET 옵션) 을 먼저 사용한 다음 설정합니다.

참고로 [openFile](#) 이후 설정한 [addSubmitParameters](#)는 [externalStorage](#)로 보고서를 재실행하면 초기화됩니다.

따라서 [externalStorage](#)로 재실행한 후에도 제출 파라미터를 설정하려면 [addSubmitParameters](#)를 다시 설정해야 합니다.

예

↳ 커스텀 핸들러로 파라미터를 전달할 때

viewer.html

```
viewer.openFile('sample.mrd', '/r/n [sample.txt]');
viewer.addSubmitParameters({'name' : 'm2soft', 'number' : '123-456'});
viewer.addSubmitParameters({'using_sample' : false});
```

customHandler.java

```
String name = request.getParameters().get("name");  
String number = request.getParameters().get("number");  
boolean using_sample = request.getParameters().get("using_sample");
```

ü 제출 요청 ID를 포함할 때

InfoTalk Server의 server.properties

```
# 요청의 고유 번호를 계약 번호(contract_num)로 설정함.  
request.identifier=contract_num
```

viewer.html

```
viewer.openFile('sample.mrd', '/rfn [sample.txt]');  
// 실패한 요청의 고유 ID인 계약 번호가 m2_20180801 일 때,  
viewer.addSubmitParameters({'contract_num' : 'm2_20180801'});
```

bind(eventName, eventHandler)

정의

bind(string eventName, function eventHandler)

설명

보고서 실행이나 제출 동작을 수행할 때 ,이벤트를 제공하기 위한 API 입니다.

인수

eventName : string

발생시킬 이벤트 종류

event :

'before-submit' : 제출버튼을 누르고 ,데이터를 서버에 전송하기 전에 이벤트 발생

'submit-finished' : 제출이 완료되면 이벤트 발생

'report-finished' : 보고서 실행이 완료되면 이벤트 발생

'lastpage' : 마지막 페이지까지 이동되면 이벤트 발생

'page-moved' : 페이지를 이동하면 이벤트 발생

'server-export' : 서버를 통하여 저장하거나 인쇄하면 이벤트 발생

'nomore-next-tab' : Tab Index 로 지정된 순서대로 탭 이동이 완료되면 이벤트 발생

eventHandler : function

발생한 event를 eventHandler 함수의 argument로 전달합니다.

event의 하위 속성을 통하여 event 결과의 상세 정보를 알 수 있습니다.

before-submit argument : { form }

form : 제출 대상 폼 객체 (제출 대상 폼 객체가 2개 이상인 경우 null)

submit-finished argument : { status, message, code, timestamp }

status : 제출 성공 여부 (true : 성공 false 실패)

message :실패일 경우 메세지

code : 실패일 경우 코드 (error / abort / timeout)

timestamp : InfoTalkServer가 비동기 방식일 때의 서버 제출 시간

report-finished argument : { totalPages, reportname, reportlongname, reportparams, subreportname }

totalPage : 전체 페이지 수

reportname : 실행 MRD 이름

reportlongname : 스크립트에서 openfile시 사용한 MRD 이름
 reportparams : 스크립트에서 openfile시 사용한 파라미터
 subreportname : 연결된 서브 리포트 MRD 이름 (경로 포함)

lastpage argument : currentPage

현재 페이지 번호이자 마지막 페이지 번호

page-moved argument : { currentPage, prevPage, formFieldModified }

currentPage : 이동한 후 페이지 번호

prevPage : 이동하기 전 페이지 번호

formFieldModified : 폼필드 편집 여부

documentTitle : 문서 제목

server-export argument : { status: true, message, action, type }

status : 서버사이드 저장 및 인쇄 결과 (true : 성공 false 실패)

message : status가 성공일 경우 저장된 파일명 ,status가 실패인 경우 에러 메시지

action : 동작 구분 ("save" : 저장 , "print" : 인쇄)

type : 저장된 파일 타입 (확장자)

주의 사항

'report-finished' 이벤트 중 subreportname기능 사용을 위해서는 /rmmffooteropt [1] 파라미터를 사용해야 합니다.

예

제출이 완료되면 이벤트가 발생하도록 하기 위한 예시입니다.

```
var eventHandler =
function(event){ if(event.status){
    alert('success : ' + event.timestamp);
}
else
    alert('fail : ' + event.message);
};
viewer.bind('submit-finished', eventHandler);
```

제출 버튼을 누르고 서버에 데이터를 보내기 전에 이벤트가 발생하는 예시 입니다.

(단 viewer.submit(), 제출 대상 폼이 두 개 이상인 경우에는 openFile 전에 이벤트를 등록 해야합니다).

```
viewer.bind('before-submit', function(form) {
// 보고서에 chk1의 체크 여부를 checked 변수에 저장
var checked = form.get("chk1").getSelected();
```

```

// 체크가 되지 않은 경우는 throw Error 처리 되는 코드
if (checked ==
  0) { alert('제출하지
    못했습니다');
  throw Error("Error"); // 콘솔 창에 Error 표시
}
else
  alert('제출 성공');

```

```

viewer.bind('before-submit', function(form) {

// 제출 대상 품 객체가 2개 이상인 경우 null
if (form === null) {
  console.log('form === null');
}
else
  var checked = form.get("chk1").getSelected();
// 체크가 되지 않은 경우 처리
if (checked ==
  0) { alert('제출하지
    못했습니다');
  throw Error("Error"); // 콘솔 창에 Error 표시
}
else
  alert('제출 성공');
}

```

보고서 로딩이 완료되면 이벤트가 발생하도록 하기 위한 예시입니다.

```

var eventHandler =
function(event){ alert('totalPage: ' +
  event.message);
};
viewer.bind('report-finished', eventHandler);

```

보고서 로딩이 완료되면 콘솔창에 mrd명 ,openfile시 사용한 mrd 명 ,사용 파라미터 ,서브 리포트 명 이벤트가 발생하도록 하기 위한 예시입니다.

```

viewer.openfile( 'http://localhost/ReportingServer/sample/mrd/
  main.mrd', '/rdata [1 @2] /rmm1footeropt [1]');
var eventHandler = function(event){
  console.log("Report Name : " + event.reportname);
  console.log("Report Long Name : " + event.reportlongname);
  console.log("Report Parameters : " + event.reportparams);
  console.log("Subreport Name : " + event.subreportname);
};
viewer.bind('report-finished', eventHandler);

```

-> console에 나타나는 결과 :

```
Report Name : main.mrd
Report Long Name : http://localhost/ReportingServer/sample/mrd/main.mrd
Report Parameters : /rdata [1@2] /rmmffooteropt [1]
Subreport Name : c:\tomcat\ReportingServer\mrd\sub.mrd
```

보고서 마지막 페이지까지 이동되면 이벤트가 발생하도록 하기 위한 예시입니다.

```
var eventHandler =
function(event){ alert(event.currentPage + 'is last
page');
};
viewer.bind('lastpage', eventHandler);
```

보고서의 페이지를 이동할 때 문서 속성의 제목을 출력하는 예시입니다.

```
var eventHandler =
function(event){ console.log('title:' +
e.documentTitle);
};
viewer.bind('page-moved', eventHandler);
```

보고서의 툴바 또는 API로 서버사이드 저장 및 인쇄 실행시 결과를 출력하는 예시입니다.

```
var eventHandler =
function(event){ if (event.status)
{
if (event.action == "save")
{
console.log("save file name : " + event.message);
console.log("save file type : " + event.type);
}
if (event.action == "print")
console.log("pdf print / file name : " + event.message);
}
else
{
console.log("error message : " + event.message);
}
};
viewer.bind('server-export', eventHandler);
```

탭 이동이 완료되었을 때 콘솔에 출력하는 예시입니다.

```
var eventHandler =
function(){ console.log("Complete");
};
viewer.bind('nomore-next-tab', eventHandler);
```

cancel()

정의

cancel()

설명

보고서 실행을 중지합니다.

예

버튼을 추가하여, 실행 중인 보고서를 중지하기 위한 예시입니다.

```
<div id="cancel">cancel</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
    var viewer = new
        m2soft.crownix.Viewer( 'http://localhost:8080/R
            eportingServer/service', 'crownix-viewer');
viewer.openFile('sample.mrd', '/rfn [sample.txt]');
$("#cancel").click(function(){
    viewer.cancel();
});
};
</script>
```


downloadFile(type)

정의

downloadFile(string type)

downloadFile(string type, function callback)

설명

현재 문서를 선택한 타입의 문서로 다운로드 합니다.

인수

type : string

저장할 파일의 타입으로 Reporting Server를 통해 저장할 수 있는 모든 타입을 지원함. 설정값 :

'pdf' , 'doc', 'xls', 'ppt', 'hwp', 'png', 'ods', 'odt'

callback : function

파일 저장시 동작할 callback 함수.

callback 변수 :

파일 다운로드 폴 경로를 전달합니다.

callback 을 다음과 같이 지정하고 사용하면 파일이 다운로드 되지 않고 경로만 전달됩니다.

```
var callback =
  function(url){ console.log(url);
};

/* 중략 */
viewer.downloadFile("pdf", callback);
```

출력 결과는 다음과 같습니다.

```
<SERVER_URL>/ReportingServer/download?crownix-client=html5viewer&
filename=2019%2F12%2F13%2F20191213182341659.pdf&rdid=20191213182341659&delete=true
```

이 때 기본 mimetype은 application/pdf이지만 mimetype을 수정하려면 다음과 같이 설정하면 됩니다.

```
var callback = function(url){
  url = url + '&mimetype=application/octet-stream';
```

```
console.log(url);
};
```

참고로 서버사이드 저장을 실행할 때 `callback`을 사용하지 않고 다운로드 된 결과를 확인하려면 [bind\(eventName, eventHandler\)](#) 에서 `server-export` 를 사용합니다.

예

버튼을 추가하여 ,실행된 보고서를 다운로드하기 위한 예시입니다.

```
<divid="download">download</div>
<divid="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
  var viewer = new
    m2soft.crownix.Viewer( 'http://localhost:8080/Re
      portingServer/service', 'crownix-viewer');
  viewer.openFile('sample.mrd', '/rfn [sample.txt]');
  $("#download").click(function(){ viewer.download
    File("pdf");
  });
};
</script>
```

getMrdPath()

정의

```
string getMrdPath()
```

설명

보고서 실행시 사용한 MRD 이름을 가져옵니다.

반환 값

mrdPath : string

보고서 실행시 사용한 MRD의 이름

예

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    var mrdList =
        [{mrdPath:'sample1.mrd', mrdParam :'/rfn [sample1.txt]'},
         {mrdPath:'sample2.mrd', mrdParam :'/rfn [sample2.txt]'}];
    viewer.openFile(mrdList);

    // 아래 결과 : ["sample.mrd", "sample2.mrd"]
    console.log(viewer.getMrdPath());
};
</script>
```

getMrdParam()

정의

string getMrdParam()

설명

보고서 실행시 사용한 MRD 파라미터를 가져옵니다.

반환 값

mrdParam : string

보고서 실행시 사용한 MRD의 파라미터

예

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
  var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
  var mrdList =
    [{mrdPath:'sample1.mrd', mrdParam :'/rfn [sample1.txt]},
     {mrdPath:'sample2.mrd', mrdParam :'/rfn [sample2.txt]}];
  viewer.openFile(mrdList);

  // 아래 결과 : ["/rfn [sample.mrd]", "/rfn [sample2.mrd]"]
  console.log(viewer.getMrdParam());
};
</script>
```

getTotalPage()

정의

number getTotalPage()

설명

총 페이지를 반환 합니다.

보고서 리포팅 중에 실행하면 ,현재까지 리포팅이 끝난 페이지 수를 반환합니다.

예

버튼을 추가하여 ,현재까지 리포팅이 끝난 총 페이지 수를 반환합니다.

```
<div id="getTotalPage">cancel</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
    var viewer = new
        m2soft.crownix.Viewer( 'http://localhost:8080/R
            eportingServer/service', 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    $("#getTotalPage").click(function(){ viewer.getTot
        alPage();
    });
};
</script>
```

print(options)

정의

print()

print(object options)

설명

실행된 보고서를 인쇄합니다.

참고로 서버사이드 인쇄를 실행할 때 결과를 확인하려면 [bind\(eventName, eventHandler\)](#) 에서 `server-export` 를 사용합니다.

인수

option : **object**

인쇄 옵션을 지정합니다.

설정 값 : { `isServerSide`, `limitedPage` }

isServerSide : *boolean*

PDF 인쇄(서버 사이드 인쇄)의 수행 여부

설정값 :

`true` : PDF 인쇄

`false` : 브라우저 인쇄 (기본)

limitedPage : *number*

브라우저 인쇄를 수행하기 위한 최대 보고서 페이지 수

설정된 페이지 수를 초과하면 `isServerSide` 값과 관계 없이 PDF 인쇄를 수행합니다.

예

버튼을 추가하여 , 보고서를 인쇄합니다.

```
<div="print">print</div>
<div="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    $("#print").click(function(){ viewer.print(); }); // 보고서 생성 후 브라우저 인쇄 수행
  };
</script>
```

인쇄 옵션을 설정하여 인쇄할 경우 아래와 같이 작성합니다.

```
// PDF 인쇄를 수행하지만 100장을 초과하면 무조건 브라우저 인쇄를 수행합니다  
viewer.print({isServerSide: true, limitedPage:100});
```

cloudPrint()

정의

cloudPrint()

설명

실행된 보고서를 클라우드 프린터로 인쇄합니다.

단 `openFile` 옵션의 `cloudPrintSpoolName` 을 지정하지 않았을 경우 기본 스푼 파일명은 "제목없음"입니다.

예

다음은 버튼을 클릭했을 때 실행된 보고서를 클라우드 프린터로 인쇄하는 스크립트 예시입니다.

```
<div id="print">print</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    $("#print").click(function(){ viewer.bind('r
    eport-finished', function(){
      viewer.cloudPrint();
    });
  });
};
</script>
```


note()

정의

note(object options)

설명

노트 기능을 사용할 수 있는 API 입니다.

인수

option : object

노트 모드 사용 여부 ,펜 종류 ,두께등을 설정할 수 있습니다. 설정

값 : { mode, tool, style }

mode : string

노트 모드 사용 여부

설정값 : { note, hide, show }

note : 노트 모드 사용

hide : 노트 숨기기

show : 노트 보이기

tool : string

노트 모드 사용시 ,사용 할 수 있는 옵션

설정값 : { pen, highlighter, text, eraser, remove, undo, redo }

pen : 펜 사용

highlighter : 형광펜 사용

text : 텍스트 노트 사용

eraser : 지우개 사용

remove : 노트 삭제

undo : 원래대로

redo : 다시설정

style : string

펜 ,형광펜의 두께 ,투명도 ,색상을 변경할 수 있는 옵션

설정값 : { width, transparency, color }

width : 두께 설정 (1 ~ 10, 18)

transparency : 투명도 설정 (0 ~ 100)

color : 색상 설정 (0 ~ 15)

[색상 인덱스 이미지]



예

버튼을 추가하여 노트를 제어합니다.

```

<input type = "button" value = "notemode1" onClick = "notemod1();" />
<input type = "button" value = "notemode2" onClick = "notemod2();" />

<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
};
var notemode1 = function() {
// 노트 기능 중 하이라이터(형광펜) 기능을 사용합니다.
viewer.note({mode : 'note', tool : 'highlighter'});
};
var notemode2 = function() {
// 노트 기능 중 하이라이터(형광펜) 기능을 사용하고 굵기는 3, 색은 빨간색 그리고 투명도를 20으로 사용합니다
viewer.note({mode : 'note', tool : 'highlighter', style:{width:3,color:8,transparency:20}});
};
</script>
    
```

cloudPrint()

정의

cloudPrint()

설명

실행된 보고서를 클라우드 프린터로 인쇄합니다.

단, `openFile` 옵션의 `cloudPrintSpoolName` 을 지정하지 않았을 경우 기본 스푼 파일명은 "제목없음"입니다.

예

다음은 버튼을 클릭했을 때, 실행된 보고서를 클라우드 프린터로 인쇄하는 스크립트 예시입니다.

```
<div id="print">print</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    $("#print").click(function(){ viewer.bind('r
    eport-finished', function(){
      viewer.cloudPrint();
    });
  });
};
</script>
```

readNfcTag()

정의

```
readNfcTag(string formId, string fieldId)
```

```
readNfcTag(function nfcdata)
```

설명

NFC 태그를 읽어 데이터를 폼필드에 삽입하는 API 입니다.

인수

formId : string

사용하려는 폼 ID

fieldId : string

데이터를 삽입하고자 하는 폼필드 ID

예

버튼을 누르고 NFC 태그를 읽어 form1의 textField1에 데이터를 삽입하는 샘플 입니다.

```
<div id="readNfcTag">readNfcTag</div>
<script>
  window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    $("#scanBarcodes").click(function(){ viewer.readN
      fctag('form1', 'textField1');
    });
  };
</script>
```

버튼을 누르고 NFC 태그를 읽어 ↓ 구분자로 데이터를 잘라 각 폼필드에 데이터를 삽입하는 샘플 입니다.

```
<div id="readNfcTag">readNfcTag</div>
<script>
  window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    $("#scanBarcodes").click(function(){
      var form1 = viewer.getForm("form1");
      viewer.readNfcTag(function(nfcData){
        var textField1 = form1.get("textField1");
```

```
var textField2 = form1.get("textField2");  
var textField3 = form1.get("textField3");  
  
textField1.setValue(nfcData[0]);  
textField2.setValue(nfcData[1]);  
textField3.setValue(nfcData[2]);  
  
});  
};  
</script>
```

scanBarcodes()

정의

```
scanBarcodes(string formId, string fieldId)
```

```
scanBarcodes(function barcodedata)
```

설명

바코드를 스캔하여 데이터를 폼필드에 삽입할 수 있는 API 입니다.

인수

formId : string

사용하려는 폼 ID

fieldId : string

데이터를 삽입하고자 하는 폼필드 ID

예

버튼을 눌러 바코드를 읽어 form1의 textField1에 데이터를 삽입하는 예제 입니다.

```
<div id="scanBarcodes">scanBarcodes</div>
<script>
window.onload = function(){
  var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
  viewer.openFile('sample.mrd', '/rfn [sample.txt]');
  $("#scanBarcodes").click(function(){ viewer.scanB
    arcodes('form1', 'textField1');
  });
};
</script>
```

버튼을 눌러 바코드를 읽고, ',' 구분자로 데이터를 잘라 각 폼필드에 데이터를 삽입하는 예제 입니다.

```
<div id="scanBarcodes">scanBarcodes</div>
<script>
window.onload = function(){
  var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
  viewer.openFile('sample.mrd', '/rfn [sample.txt]');
  $("#scanBarcodes").click(function(){
    viewer.scanBarcodes(function(barcodeData){
      var form1 = viewer.getForm("form1");
      var token = barcodeData.split(',');

      var textField1 = form1.get("textField1");
    });
  });
};
</script>
```

```
var textField2 = form1.get("textField2");  
var textField3 = form1.get("textField3");  
  
textField1.setValue(token[0]);  
textField2.setValue(token[1]);  
textField3.setValue(token[2]);  
  
});  
};  
</script>
```

setAlterFontMap

정의

```
setAlterFontMap(object fontMap)
```

설명

대체 폰트 적용을 위한 폰트 맵을 지정할 수 있는 API 입니다.

인수

fontMap : **object**

설정값 : { fontName, alternativeFontName }

fontName : *string*

대체 폰트를 적용할 폰트명

alternativeFontName : *string*

대체 폰트명

예

```
<script>
window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.setAlterFontMap({'HY헤드라인M' : 'hyheadline medium'});
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
};
</script>
```


setDownloadFileName(fileName)

정의

setDownloadFileName(string fileName)

설명

현재 문서를 다운로드하기 전에 저장할 이름으로 지정 합니다.

(단, 한글파일명은 지정이 되지 않습니다.)

인수

fileName : string

저장할 파일의 이름

예

버튼을 추가하여 실행된 보고서를 저장하기 전에 저장할 이름인 **test**로 지정하는 예제입니다.

```
<div id="setDownloadFileName">setDownloadFileName</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
    $("#setDownloadFileName").click(function(){ viewer
        er.setDownloadFileName('test');
    });
};
</script>
```

setStateLessMode(isStateless)

정의

setStateLessMode(boolean isStateless)

설명

뷰어와 서버간의 페이지 전송 방식을 지정합니다.

Stateful 모드에서는, 서버에 보고서를 저장하며, 페이지를 이동할 때 서버에게 해당 페이지를 요청합니다.

Stateless 모드에서는, 클라이언트에 보고서를 저장하며, 페이지를 이동할 때 서버에게 요청하지 않습니다. 이

API를 사용하지 않으면 **Stateless** 모드로 동작합니다.

인수

isStateless : **boolean**

설정값 :

true: 전체 페이지를 내려 받습니다 (기본값)

false: 이동하는 페이지만 내려 받습니다.

예

```
<script>
window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.setStateLessMode(false);
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
};
</script>
```

submit(url, datasource, servicecode, target, [csvfileName])

정의

```
submit(string url, string datasource, string servicecode, string target)
```

```
submit(string url, string datasource, string servicecode, string target, string csvFileName)
```

설명

보고서에 제출 버튼을 넣지 않고 API로 제출합니다.

인수

url : string

폼필드의 데이터가 전송되는 InfoTalk Server의 URL주소

datasource : string

InfoTalk Server에 등록되어 있는 데이터 소스 중 사용할 데이터

참고 사항 : 폼필드 속성 중 ,제출과 관련된 속성 에서 [데이터 소스]의 내용을 참고하세요

servicecode : string

제출 방식을 선택적으로 사용할 수 있도록 서비스 코드

참고 사항 : 폼필드 속성 중 ,제출과 관련된 속성 에서 [서비스 코드]의 내용을 참고하세요

target : string

제출할 품 ID 리스트(콤마,)로 구분)

전체 제출시에는 빈 문자열

참고 사항 : 폼필드 속성 중 ,제출과 관련된 속성 에서 [제출 대상 품]의 내용을 참고하세요

csvfileName : string

CSV 파일 제출 방식일때 ,CSV 파일 이름을 설정하는 옵션

CSV 파일 제출이 아닐시에는 빈 문자열

예]

```
<script>
  var viewer;
  window.onload = function(){
    viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
  };
  function
  submit(){ viewer.submit(
    'http://localhost:8080/InfoTalkServer/service', ||
    'datasource', |
    '1300', |
    ');
  };
</script>
<input type="button" value="submit" onclick="javascript:submit();" />
```

useAdjustLetterSpace(use)

정의

useAdjustLetterSpace(boolean use)

설명

자간 조정을 통한 문서 보정 허용 여부에 대한 설정입니다.

사용하지 않을 경우 페이지의 로딩은 빠르지만, 문서의 레이아웃이 맞지 않을 수 있습니다. 이

API를 사용하지 않으면 기본적으로 자간 보정을 수행합니다.

인수

use : **boolean**

자간 조정을 통한 보정을 허용 여부

설정값 : true (기본값), false

주의 사항

openFile 함수를 실행하기 전에 설정해야 합니다.

예

```
<script>
window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.useAdjustLetterSpace(false);
    viewer.openFile('sample.mrd', '/r/n [sample.txt]');
};
</script>
```

관련 API

[useAdjustCharacterScale\(use\)](#)

useAdjustCharacterScale(use)

정의

useAdjustCharacterScale(boolean use)

설명

장평 조정을 통한 문서 보정 허용 여부에 대한 설정입니다.

사용하지 않을 경우 페이지의 로딩은 빠르지만, 문서의 레이아웃이 맞지 않을 수 있습니다.

인수

use : boolean

장평 조정을 통한 보정 허용 여부

설정값 : true, false (기본값)

주의 사항

openFile 함수를 실행하기 전에 설정해야 합니다.

예

```
<script>
window.onload = function(){
  var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
  viewer.useAdjustCharacterScale(false);
  viewer.openFile('sample.mrd', '/rfn [sample.txt]');
};
</script>
```

관련 API

[useAdjustLetterSpace\(use\)](#)

useWebFont(use)

정의

useWebFont(boolean use, object fontMap)

설명

웹 폰트 사용 여부를 설정합니다.

이 API를 사용하지 않으면 웹 폰트를 사용하지 않습니다.

또한 URL Encoding, File Encoding, Web Server의 확장자 제한 등의 문제로 웹 폰트를 사용할 수 없을 때, fontMap을 설정하여 웹 폰트를 사용할 수 있습니다.

인수

use : boolean

웹 폰트를 사용 여부

설정값 : true, false (기본값)

fontMap : object

설정값 : { fontName, fontFileName }

fontName : string

폰트 이름, 디자이너에 표기되는 폰트 이름과 동일

fontFileName : string

폰트 파일명 ReportingServer/font 디렉토리에 있는 파일명과 동일 (대소문자 포함)

주의 사항

openFile 함수를 실행하기 전에 설정해야 합니다.

예

기본적인 웹 폰트 사용

```
<script>
window.onload = function(){
  var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
  viewer.useWebFont(true);
  viewer.openFile('sample.mrd', '/rfn [sample.txt]');
};
</script>
```

fontMap을 설정하여 웹 폰트를 사용

```
<script>
window.onload = function(){
  var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
  var fontMap = {
    '맑은 고딕':'malgun',
    '돋움?' 'dotum',
    '휴먼등근헤드라인 'HMKMRHD'
  };
  viewer.useWebFont(true, fontMap);
  viewer.openFile('sample.mrd', '/rfn [sample.txt]');
};
</script>
```


다이얼로그 관련 API

툴바에서 제공하는 다이얼로그를 API로 호출하는 기능에 대하여 설명합니다.

showFormDataStorageDialog()

정의

showFormDataStorageDialog()

설명

폼 편집 모드로 실행된 보고서에서 입력 중의 폼 데이터를 저장하기 위한 다이얼로그 박스를 호출하는 기능입니다.

이 기능은 툴바 버튼의 '데이터' 버튼을 누르면 호출하고 있지만 API에서도 호출할 수 있습니다.

주의 사항

이 기능은 [Storage 기능이 포함된 viewer 모듈](#)을 사용해야 정상 동작합니다.

예

```

<input type="button"
  value="showFormDataStorageDialog"
  onclick="javascript:showDialog()" >
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    function showDialog(){
      viewer.showFormDataStorageDialog();
    };
  };
</script>

```

showMMLStorageDialog()

정의

showMMLStorageDialog()

설명

실행된 보고서를 MML 형식으로 저장하기 위한 다이얼로그 박스를 호출하는 기능입니다.
이 기능은 툴바 버튼의 **보고서** 버튼을 누르면 호출하고 있지만 API에서도 호출할 수 있습니다.



주의 사항

이 기능은 [Storage 기능이 포함된 viewer 모듈](#)을 사용해야 정상 동작합니다.

예

```
<input type="button"
  value="showMMLStorageDialog"
  onclick="javascript:showDialog()" >
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    function showDialog(){
      viewer.showMMLStorageDialog();
    };
  };
</script>
```

showInqueryDialog()

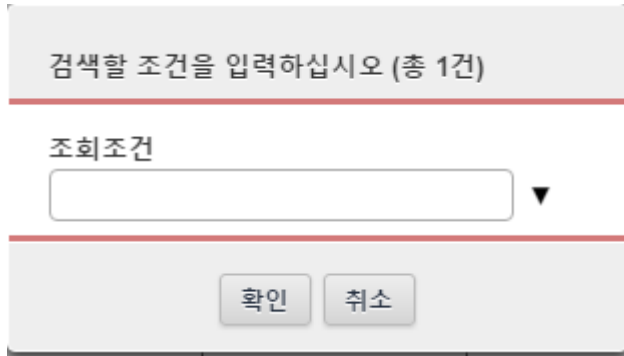
정의

showInqueryDialog()

설명

실행된 보고서의 조회 툴바 다이얼로그 박스를 호출하는 기능입니다.

이 기능은 툴바 버튼의 조회툴바 버튼을 눌러서 호출 할 수도 있고 API에서도 호출할 수 있습니다.



주의 사항

이 기능은 조회툴바가 적용된 보고서를 실행하고 사용해야 정상 동작합니다.

예

```

<input type="button"
  value="showInqueryDialog"
  onclick="javascript:showDialog()" >
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    function showDialog(){
      viewer.showInqueryDialog();
    };
  };
</script>

```

showMovePageDialog()

정의

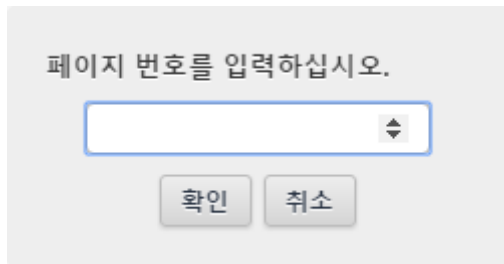
showMovePageDialog()

설명

실행된 보고서의 페이지 이동 다이얼로그 박스를 호출하는 기능입니다.

페이지 이동 다이얼로그가 뜨며, 다이얼로그에 입력한 페이지 번호로 이동합니다.

생성되지 않은 페이지 또는 첫 페이지 이전으로 이동하려고 하면, 경고 메시지 창이 나타납니다. 이 기능은 툴바 버튼의 페이지 이동을 누르면 호출하고 있지만, API에서도 호출할 수 있습니다.



주의 사항

이 기능은 조회툴바가 적용된 보고서를 실행하고 사용해야 정상 동작합니다.

예

버튼을 추가하여, 실행된 보고서 페이지 이동 다이얼로그를 띄우는 예시입니다.

```
<input type="button"
  value="showMovePageDialog"
  onclick="javascript:showDialog()" >
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    function showDialog(){
      viewer.showMovePageDialog();
    };
  };
</script>
```

Run-time 보고서 관련 API

런타임 리포트를 사용하는 보고서에서 사용하는 API에 대하여 설명합니다.

addColumn(name, width, cellMergeOption)

정의

addColumn(string name, number width, number cellMergeOption)

설명

런타임 리포트를 사용하는 보고서에서 실행할 데이터 매핑 정보를 저장하는 API입니다 반드시 [addColumn](#) API와 함께 사용합니다.

파라미터

name : string

열(컬럼)을 추가할 표 객체의 첫 번째 셀 필드명

width : number

열(컬럼)의 너비

cellMergeOption : number

셀 병합관련 옵션

데이터 셋 연결 대화상자의 셀 병합관련 옵션을 지정합니다.

option 값은 다음 중 한 개 이거나 여러 개를 조합(논리적 OR 연산)하여 사용할 수 있습니다.

설정값 :

1 : 표의 왼쪽 x 좌표 고정

2 : 표의 오른쪽 x좌표 고정

예

표에서 필드명이 반복1, 너비가 150, 그리고 만들었던 표의 왼쪽 x좌표 고정인 열(컬럼)을 추가하는 예제 입니다.

```
var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
viewer.openFile('sample.mrd', '/rfn [sample.txt]');

viewer.addColumn('repeat', 150, 1)
  .addCell(1, 'CompanyName', "", "", "", 0, "")
  .addCell(2, 'COMPANY_NAME_sub1', "", "", "", 1, "");

viewer.openFile('sample.mrd', '/rp [param1] [param2]');
```

addCell(index, title, variable, summarize, command, attribute, option, refField)

정의

addCell(number index, string title,string variable,string summarize,string command,string attribute, number option, string refField)

설명

런타임 리포트를 사용하는 보고서에서 실행할 데이터 매핑 정보를 저장하는 API입니다. 반드시 [addColumn](#) API와 함께 사용합니다.

파라미터

index : number

셀 인덱스, 1부터 시작

title : string

셀에 표시 할 텍스트 내용

variable : string

필드명 데이터 셋 연결 대화상자 필드명 변수이름을 지정

summarize : string

요약작성 데이터 셋 연결 대화상자 요약작성 요약작성 내용 지정

command : string

수행코드 CELL속성을 바꾸기 위한 코드를 나타내는 문자열

설정 값 :

'HA' : 가로 정렬

'VA' : 세로 정렬

" : 중앙 정렬 (기본 값)

attribute : string

수행코드에 대한 속성값

설정 값 :

'1' :

HA, 1 => 왼쪽 정렬

VA, 1 => 상단 정렬

'2' :

HA, 2 => 오른쪽 정렬

VA, 2 => 중앙 정렬

'3' :

VA, 3 => 하단 정렬

기본 값 : 중앙 정렬

option : number

셀 병합 옵션 데이터 셋 연결 대화상자의 셀 병합 옵션 지정.

설정값 :

0 : 셀 병합 하지 않음

1 : 셀 병합

2 : 모두 병합

4 : 옆 셀 병합

8 : 한번만 쓰기

refField : string

참조 필드 명

데이터 셋 연결 대화상자의 참조 필드 리스트 지정

필드명이 하나 이상인 경우 @ 문자로 구분하여 여러 개의 필드명을 지정합니다.

예

addColumn으로 추가된 열(컬럼)의 첫 번째 셀에는 회사명 으로 텍스트를 표시 ,가로 정렬을 왼쪽 정렬로 하고 셀 병합은 하지 않는 셀을 추가합니다.

두 번째 셀에는 COMPANY_NAME_sbu1로 필드를 지정하고 ,셀 병합 옵션을 가진 셀을 추가하는 예제입니다.

```
var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');

viewer.addColumn("반복1", 150, 1)
    .addCell(1, '회사명', "');
```

```
.addCell(2, ", 'COMPANY_NAME_sub1', ", 'HA', '1', 0, ");  
  
viewer.addColumn('반복2', 100, 1)  
.addCell(1, '부서명 *****', "  
.addCell(2, ", 'DEPT_NAME_sub1', ", ", ", 1, ");  
  
viewer.openFile('sample.mrd', '/rfn [sample.txt]');
```

하이퍼링크 관련 API

하이퍼링크 기능을 사용하는 보고서에서 사용하는 API에 대하여 설명합니다.

nextDoc()

분류

viewer

정의

nextDoc()

설명

하이퍼링크로 연결된 보고서에서 연결된 보고서로 이동하기 위한 API 입니다.

예

```
<div id="nextDoc">nextDoc</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');

    $("#nextDoc").click(function(){ viewer
        er.nextDoc();
    });
};
</script>
```

preDoc()

분류

viewer

정의

preDoc()

설명

하이퍼링크로 연결된 보고서에서 툴바 버튼이나 [nextDoc\(\)](#) API로 이동한 후, 이동하기 이전의 보고서로 돌아가기 위한 API입니다.

예

```
<div id="prevDoc">prevDoc</div>
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');

    $("#prevDoc").click(function(){ viewer
er.prevDoc();
});
};
</script>
```

M2soft.crownix.Layout의 API

M2soft.crownix.Layout 클래스에서 제공하는 API에 대하여 설명합니다. 포함된 API를 이용하여 뷰어의 설정을 직접 바꿀 수 있습니다.

Theme

뷰어의 색상 테마를 정의하기 위한 API를 제공합니다.

[setTheme\(theme\)](#)

[setTheme\(option\)](#)

setTheme(theme)

정의

setTheme(string theme)

설명

뷰어의 테마색상을 설정합니다.

인수

theme : string

적용 가능한 테마 색상은 아래와 같습니다. 대소문자는 구별하지 않습니다.

설정 값 : 'Gray' (기본 값), 'White', 'Black', 'Blue', 'Pink', 'Orange', 'Mint', 'Green', 'Skyblue',
'Transparent' (투명)

예

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  m2soft.crownix.Layout.setTheme('Blue');
  window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
  };
</script>
```


setTheme(customTheme)

정의

setTheme(object customTheme)

설명

뷰어의 사용자 정의 색상을 설정합니다.

툴바와 아이콘 및 보고서 배경의 색상을 자유롭게 설정할 수 있습니다.

인수

customTheme : **object**

사용자가 직접 색상을 지정하여 뷰어의 배경 색상과 투바 색상 및 아이콘 효과 등을 설정할 수 있습니다.

적용 가능한 항목은 아래와 같습니다.

설정 값 : { toolbarColor, toolbarIconColor, toolbarWidth, toolbarEffectColor, containerColor, pageBorderColor, pageBorderWidth }

toolbarColor : *string*

툴바의 배경색

toolbarIconColor : *string*

툴바의 아이콘 색상

toolbarWidth : *string*

툴바의 두께 (단위 : px)

최소 값 : '44px'

toolbarEffectColor : *string*

툴바의 아이콘의 border, hover, active 색상

containerColor : *string*

뷰어의 배경색

pageBorderColor : *string*

페이지 테두리의 색상

pageBorderWidth : *string*

페이지 테두리 두께 (단위 : px)

예

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  m2soft.crownix.Layout.setTheme({
    toolbarColor: '#E0E0E0',
    toolbarIconColor: '#4D4D4D',
    toolbarEffectColor: '#333333',
    toolbarWidth: '44px',
    containerColor: '#F0F0F0',
    pageBorderColor: '#E0E0E0',
    pageBorderWidth: '1px'});

  window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
  };
</script>
```

M2soft.crownix.Resource의 API

M2soft.crownix.Resource 클래스에서 제공하는 API에 대하여 설명합니다.

포함된 API를 이용하여 뷰어의 설정을 직접 바꿀 수 있습니다.

setLanguage(language)

정의

setLanguage(string language)

설명

HTML5 뷰어의 리소스 언어를 설정합니다. 반드시 `viewer` 객체를 생성하기 전에 사용해야 합니다.

인수

language : string

설정 언어

설정 값 :

'ko' : 한국어 (기본값)

'en' : 영어

'ja' : 일본어

예

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
window.onload = function(){
    m2soft.crownix.Resource.setLanguage('en');
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
};
</script>
```

M2soft.crownix.util.NumberUtil의 API

M2soft.crownix.util 클래스에서 제공하는 API에 대하여 설명합니다.

포함된 API를 이용하여 뷰어의 설정을 직접 바꿀 수 있습니다.

setLocaleFormat(locale, format)

정의

setLocaleFormat(string locale)

setLocaleFormat(string locale, object format)

설명

HTML5 뷰어의 숫자 포맷을 설정합니다. 반드시 `Viewer`

객체를 생성하기 전에 사용해야 합니다.

인수

locale : *string*

설정 언어

설정 값 :

'ko' : 한국어 (기본값)

'en' : 영어

'ja' : 일본어

'zh' : 중국어 `-format` 을 사용할 경우만

format : *object*

서식 설정

설정 값 : { `currencyFormat`, `numberFormat` }

currencyFormat : *string*

통화 포맷

numberFormat : *string*

숫자 포맷

예

₩ 단어를 영어로 변환 후 통화 포맷을 \$로 표시

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    m2soft.crownix.util.NumberUtil.setLocaleFormat('en');
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
  };
</script>
```

₩ 단어를 ¥로 변환 후 통화 포맷을 지정

```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;"></div>
<script>
  window.onload = function(){
    m2soft.crownix.util.NumberUtil.setLocaleFormat('j
    a,
    {currentFormat : '#,#JPY', numberFormat : '#}');
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.openFile('sample.mrd', '/rfn [sample.txt]');
  };
</script>
```

M2soft.crownix.Util의 API

M2soft.crownix.Util 클래스에서 제공하는 API에 대하여 설명합니다.

포함된 API를 이용하여 뷰어의 설정을 직접 바꿀 수 있습니다.

ActionEvents

정의

Action Events = { start, move, end, out, over};

설명

HTML5뷰어는 브라우저의 이벤트를 통해서 마우스, 펜, 터치 이벤트를 사용합니다.

브라우저의 종류에 따라 여러 이벤트를 제공하는 경우가 많아서 혼용해서 쓰는 경우도 많습니다.

또한 특수한 상황이나 여러 이벤트 중 특정 이벤트만 적용하려고 할 수도 있습니다.

ActionEvent는 이러한 상황에서 이벤트를 재정의하기 위한 API입니다.

재정의 할 이벤트는 사용할 브라우저에서 제공하는 이벤트로만 설정할 수 있습니다. 예를 들어 터치가 지원되지 않는 브라우저에서 터치로 재정의하면 동작하지 않습니다.

다음의 내용을 참조하여 사용할 수 있는 인수를 설정해야 합니다.

https://developer.mozilla.org/en-US/docs/Web/API/Pointer_events#Event_types_and_Global_Event_Handlers

인수

start : string

누르는 이벤트

설정 값 예시 : 'pointerdown' | 'mousedown'

move : string

이동 이벤트

설정 값 예시 : 'pointermove' | 'mousemove'

end : string

동작 종료 이벤트

설정 값 예시 : 'pointerup' | 'mouseup'

out : string

포인터가 요소에서 벗어날 때의 이벤트

설정 값 예시 : 'pointerout' | 'mouseout'

over : string

포인터가 요소에 진입할 때의 이벤트

설정 값 예시 : 'pointerover' | 'mouseout'

예

```
<script>
var viewer;
window.onload = function(){
viewer = new m2soft.crownix.Viewer('<SERVER_URL>', 'crownix-viewer');
viewer.openFile(
'sample.mrd', '/rfn [sample.txt] /rformmode [2]',{mirroring: {enable: true}});
m2soft.crownix.Util.ActionEvents = {
start: 'mousedown',
move: 'mousemove',
end: 'mouseup',
out: 'mouseout',
over: 'mouseover'
};
};
</script>
```

m2soft.crownix.FormDialog의 API

뷰어에서 제공하는 다이얼로그를 API로 작성하는 기능에 대하여 설명합니다.

show(contents)

정의

show(string contents)

show(string contents, object options)

설명

보고서에서 다이얼로그를 생성하기 위한 API입니다.

인수

contents : string

다이얼로그에 나타날 메시지

options : object

다이얼로그를 나타낼 때 적용할 수 있는 옵션

설정 값 : { type, position, title, ok, cancel, time, enableClose, outsideClose, switchConfirmButton, okHandler, cancelHandler, closeHandler }

type : string

다이얼로그 박스 종류

'alert' : 알람 메시지 창 (기본값)

'toast' : 토스트 메시지 창

잠깐 나타났다가 일정 시간 후 종료하는 메시지 창

'confirm' : 확인, 취소 버튼이 있는 메시지 창

position : string

다이얼로그 박스의 팝업 위치

'center' : 중앙 (기본값)

'top' : 화면 상단

'bottom' : 화면 하단

title : string

다이얼로그 박스의 제목

type이 alert' 또는 confirm' 인 경우 사용 가능

ok : string

다이얼로그 박스의 확인 버튼의 대체 문구

`type`이 'alert' 또는 'confirm' 인 경우 사용 가능

`cancel` : *string*

다이얼로그 박스의 취소 버튼의 대체 문구

`type`이 'confirm' 인 경우 사용 가능

`time` : *number*

메시지 창이 종료되는 시간 (초 단위)

`type`이 'toast' 인 경우 사용 가능 기본

값 : 5초

`enableClose` : *boolean*

alert 창과 confirm 창에 닫기 x 버튼 표시

기본 값 : false

`outsideClose` : *boolean*

토스트 메시지를 `close()`의 외부 API 또는 `esc` 키로 닫는 것을 허용

`type`이 'alert' 또는 'confirm' 인 경우 사용 가능

기본 값 : true

`switchConfirmButton` : *boolean*

다이얼로그 박스의 확인버튼과 취소 버튼의 순서 변경

`type`이 'alert' 또는 'confirm' 인 경우 사용 가능

기본 값 : false

`okHandler` : *function*

메시지 창에서 확인 버튼이 눌렸을 때 실행시킬 함수

- `type`이 'alert' / 'confirm'의 경우 사용 가능

`cancelHandler` : *function*

메시지 창에서 취소 버튼이 눌렸을 때 실행시킬 함수

- `type`이 'confirm'일 경우 사용 가능

`closeHandler` : *function*

사용자가 메시지 창을 닫을 때 실행시킬 함수

메시지 창을 닫는 방법 :

'alert' / 'confirm' 창의 x 버튼으로 종료

'toast' 창에서 여백을 눌러 종료

예

3종류의 타입의 다이얼로그 창을 띄우는 예시입니다.

```
<input type="button" id="alert" value="alert message" />
<input type="button" id="toast" value="toast message" />
<input type="button" id="confirm" value="confirm message" />

<div id="crownix-viewer" style="position:absolute;width:100%;height:100%"></div>
<script>
window.onload = function(){
    var viewer = new m2soft.crownix.Viewer(server_url, "crownix-viewer");
    viewer.openFile("sample.mrd", "/ rfn [sample.txt]");
    var dialog = new m2soft.crownix.FormDialog("#crownix-viewer");
    $("#alert").click(function(){
        dialog.show("alert message!!"); // alert 메시지 창 -기본
    });
    $("#toast").click(function(){ dia
        log.show(
            "toast message!!",
            {
                type : "toast",    // toast 메시지 창
                position : "bottom", // 화면 중앙에 팝업
                time : 1,          // 팝업되고 1초뒤 종료
                closeHandler : function(){
                    console.log("closed."); //종료시 콘솔창에 메시지 작성
                }
            }
        );
    });
    $("#confirm").click(function(){
        dialog.show(
            "confirm message!!",
            {
                type : "confirm",    // confirm 메시지 창
                //ok 버튼이나 cancel 버튼 클릭시 콘솔창에 메시지 작성
                okHandler : function(){ console.log("ok");},
                cancelHandler : function(){ console.log("cancel");}
            }
        );
    });
};
</script>
```

HTML5 Viewer Callback Function 사용법

M2soft.crownix.shareManager 클래스에서 제공하는 API에 대하여 설명합니다.

포함된 API를 이용하여 뷰어의 설정을 직접 바꿀 수 있습니다.

HLinkClicked(target, url, params)

정의

HLinkClicked(string target, string url, string params)

설명

HTML5 뷰어에서 하이퍼링크 객체를 마우스 클릭했을 때 발생하는 이벤트입니다.

반드시 `viewer` 객체를 생성하기 전에 사용해야 합니다.

인수

target : string

하이퍼링크 속성 대화상자에서 클릭 시 이벤트 발생 체크박스 옆의 텍스트 박스에 입력한 문자열 이
문자열은 체크박스가 선택되었을 때만 유효
(단 HTML5 뷰어에서 인수로 사용 안함)

url : string

연결 대상

params : string

연결시 사용할 파라미터 (RD문서인 경우)

예

ü 인수와 보고서의 하이퍼링크 속성과의 관계

HLinkClicked

1. target:HLinkClicked
2. url:a.mrd
3. params:/rfn [a.txt] /rp [b.txt]

```

<script>
function HLinkClicked(Target, URL, Params) {
    var overlayViewer = new m2soft.crownix.Viewer(server_url);
    overlayViewer.openFile(URL, Params);
};

window.onload = function(){
    m2soft.crownix.Viewer.debug = true;
    var viewer = new m2soft.crownix.Viewer(server_url, 'crownix-viewer');
    viewer.openFile('h5_overlay.mrd', '/rdata ["]');
};
</script>

```

보고서 작성 가이드

보고서가 실행되는 속도를 더 빠르게 할 수 있는 옵션이나 MRD 템플릿 작성 요령을 설명합니다.

성능 관련 옵션

보고서가 실행되는 속도를 더 빠르게 할 수 있는 옵션에 대하여 설명합니다.

• 텍스트 자동 보정

브라우저마다 텍스트를 표시하는 계산 방식이 조금씩 다릅니다.

그러므로, 텍스트의 사이즈가 동일하더라도 브라우저에 따라 다르게 보일 수 있습니다.

텍스트 자동 보정 기능을 사용하여 이 문제를 해결할 수 있습니다.

텍스트 자동 보정이란 텍스트가 정해진 셀 영역을 벗어나면 셀 영역 내에 채워질 때까지 자간(letter-spacing)을 조금씩 조정하여 셀 안에 채워지게 하는 기능입니다.

[useAdjustLetterSpace](#) API를 이용해 텍스트 자동 보정을 허용할지 말지 설정할 수 있습니다.

텍스트 자동 보정을 허용하지 않으면, 텍스트 길이를 계산하는 시간을 절약할 수 있기 때문에 보고서 생성 속도가 빨라집니다.

텍스트 자동 보정을 하지 않고도 모든 환경에서 동일하게 텍스트를 표시하려면, 템플릿을 작성할 때 가급 적 셀 내에 텍스트가 꼭 채워지지 않게 여백을 충분히 설정하는 것이 좋습니다.

• 페이지 단위 전송

여러 페이지로 이루어진 보고서를 페이지 단위로 전송하면 전체 페이지를 한번에 받는 방식보다 최초 실행 속도가 빠릅니다.

그러나 페이지를 이동할 때마다 해당 페이지를 서버서부터 전송 받기 때문에 페이지 이동 시 화면에 표시 하는 속도는 느려질 수 있습니다.

[setStateLessMode](#) API를 이용해 페이지 단위로 전송할지 말지 설정할 수 있습니다.

전송방식에 대한 자세한 설명은 [Stateful & Stateless 모드](#)를 참조하시기 바랍니다.

• 이미지 인덱싱

보고서 내의 여러 페이지에 동일하게 표시되는 이미지에 대해 인덱스 처리를 하여 동일한 이미지를 매번 생성하는 것을 방지하여, 시스템의 메모리 사용량을 줄이고, 보고서 생성 속도를 개선할 수 있습니다.

`/rimindexing` 파라미터를 사용할 때 이미지 인덱싱을 합니다.

파라미터에 대한 자세한 설명은 [Crownix Report 매뉴얼](#)의 관련 파라미터 항목을 참조하시기 바랍니다.

● 다량의 텍스트가 포함된 보고서

보고서에 다량의 텍스트가 포함되면 Openfile시 텍스트 각각에 HTML 속성 및 폰트 계산 작업으로 인해 속도가 느려지게 됩니다.

openFile [TextOnCanvas](#) 옵션을 사용하면 보고서를 작성할 때 보고서의 각 텍스트에 HTML 속성을 부여하지 않고 하나의 Canvas 이미지로 생성하여 보고서 렌더링 속도가 빨라질 수 있습니다.

(보고서 렌더링은 빨라지지만 텍스트가 이미지이기 때문에 확대나 축소를 하면 흐려질 수 있습니다.)

● 브라우저 별 Canvas 크기 제한

HTML5 Viewer는 브라우저 마다 한 페이지에 그릴 수 있는 Canvas 크기에 제한이 있습니다. 예를 들어 Chrome 브라우저에서 보고서 30페이지 이상의 내용을 one page 보고서로 작성한 보고서를 Chrome 브라우저에서 열면 보고서가 그려지지 않습니다.

다음은 A4 용지 크기 기준으로 one page 보고서(/ronepgrpt 적용)로 테스트 한 결과입니다.

브라우저	Canvas Size (pixels) / 페이지 수
Chrome	32K / 29p
Firefox	32K / 29p
Safari	117K / 105p
Safari Mobile	19K / 17p
Opera	32K / 29p
IE 11	16K / 14p

● 보고서 스크롤 이벤트

[openFile](#)의 [scrollOption](#)을 통해 HTML5 Viewer 스크롤 이벤트(데스크탑의 경우 클릭 이벤트, 모바일의 경우 터치 이벤트)를 제어할 수 있습니다.

ü 주의사항 : 데스크탑 환경에서 [scrollOption](#)에 적용할 옵션은 브라우저마다 다르게 설정해야 합니다.

모바일의 경우 `disableTouch`만 설정해도 스크롤 이벤트를 제어할 수 있습니다.

그러나 데스크탑에서는 브라우저마다 이벤트 처리가 달라 다음과 같은 조합으로 동시에 옵션을 적용해야 스크롤 이벤트를 제어할 수 있습니다.

브라우저	동시에 적용해야 하는 옵션
Chrome	<code>disablePointer</code> <code>disableMouse</code>
Safari	<code>disableTouch</code>
IE 11	<code>disableMouse</code>
모든 환경	<code>disableTouch</code> <code>disablePointer</code> <code>disableMouse</code>

- 텍스트 상자 속성의 단어 단위로 문장 나눔 속성

디자이너에서 작성된 보고서의 폰트는 pt 단위이며 브라우저에서는 단위 환산을 하며 보고서가 실행되는 브라우저에 따라 환산 공식이 다릅니다.

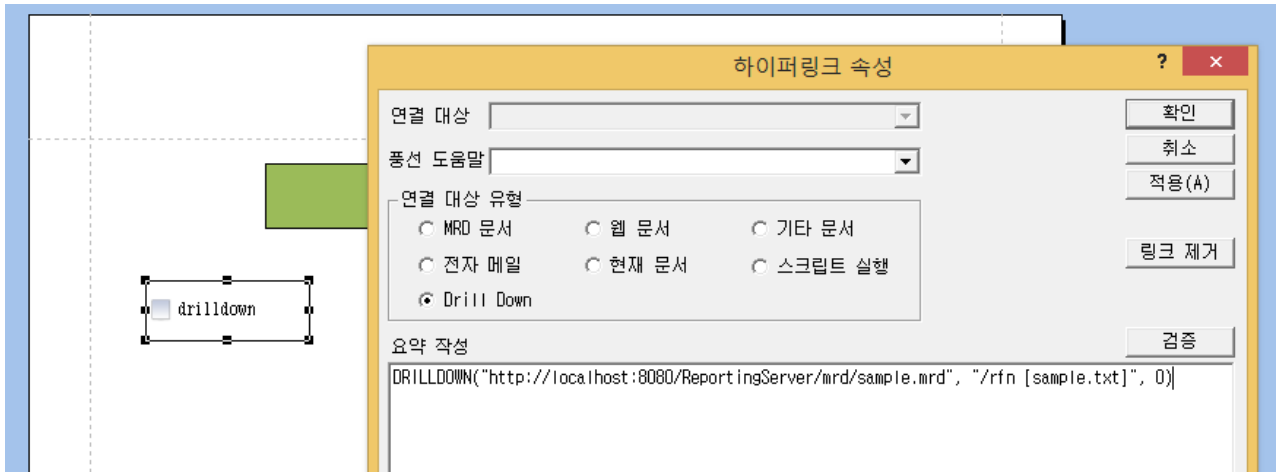
또한 브라우저에 따라 공란 (스페이스) 을 많이 사용하면 일부 개수를 무시하여 표현합니다.

이러한 동작을 보완하기 위하여 텍스트 상자 속성의 HTML5 Viewer 실행 속성 에서 [단어 단위로 문장 나눔] 설정을 사용할 수 있습니다.

이 동작은 문장을 단어 단위로 잘라서 연산해야하기 때문에 보고서 실행 성능이 저하됩니다.

드릴다운 보고서 작성 팁

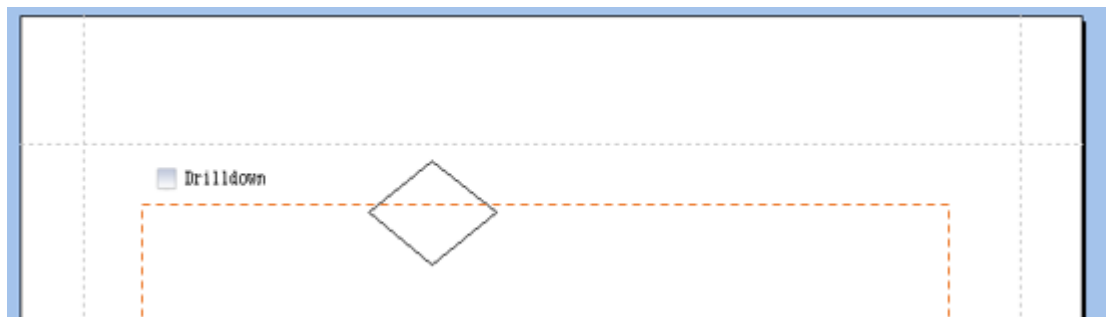
드릴다운 보고서는 보고서 작성시 객체에 [맞추기]-[하이퍼링크 속성]에서 **drilldown**을 선택하여 하이퍼링크 기능을 적용하고 조건에 따라 이 객체 밑이나 옆으로 다른 보고서가 나타나도록 하는 보고서입니다. 드릴다운은 아래로 펼쳐지면서 기존 객체들을 아래로 미는 것이 기본 방식이지만 옵션에 따라 오른쪽으로 펼쳐지는 드릴 라이트도 있으며 아래쪽 방향으로 펼쳐져 객체를 밀지 않고 오버로 띄우는 드릴 오버 방식이 있습니다.



• 드릴다운 보고서 제약사항

아래와 같이 'drilldown' 체크박스를 선택해서 밑에 보고서가 펼쳐도록 할 경우, 펼쳐 보여야 하는 영역에 겹치는 다른 객체가 있으면 드릴다운이 동작하지 않습니다.

레이어 기능과 함께 사용할 수 없습니다.



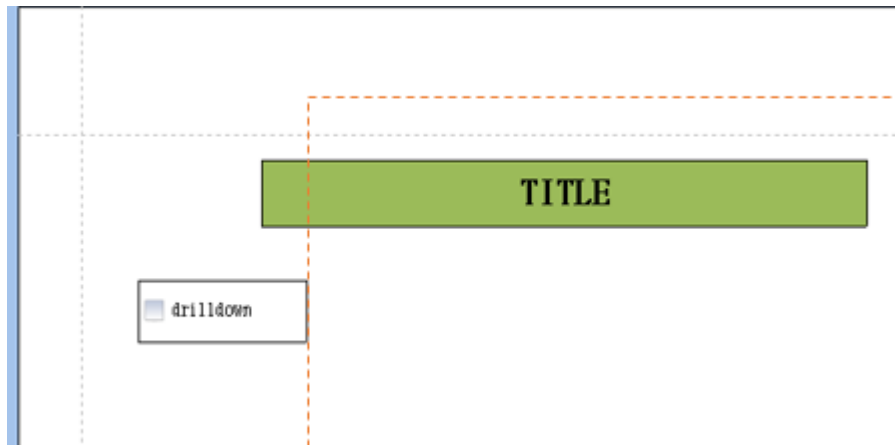
- 드릴다운 보고서의 제약사항을 무시하려면?

펼쳐지는 영역과 겹치지만 꼭 겹쳐지는 객체를 보여주고 싶다고 해서 아래와 같이 보고서를 작성하면 드릴다운이 실행되지 않습니다.

왜냐하면 Title 부분이 드릴다운해서 펼쳐져야하는 보고서 영역과 겹치기 때문입니다.

이 Title을 고정적으로 보고서에 나타나게 하고 싶다면 Title 부분을 다른 MRD로 만들고 이 Title MRD를 드릴 오버로 실행시키도록 합니다.

이 Title을 드릴오버로 호출하는 스크립트는 보고서 실행이 완료되고 바로 호출해주면 되기 때문에 `onloadComplete` 스크립트에서 Title 부분을 드릴오버로 펼치도록 합니다.



제약 사항

HTML5 Viewer의 제약 사항에 대하여 설명합니다.

지원하지 않는 기능

Crownix Report Designer에서 적용한 기능 중 HTML5 Viewer에서 지원하지 않는 기능을 설명합니다.

지원하지 않는 기능을 사용하여 작성한 보고서는 실행하는 환경에 따라 출력 결과가 달라집니다.

예를 들면 Crownix Report ActiveX Viewer에서는 표시되는 객체가 HTML5 Viewer에서는 보이지 않거나 다른 모양으로 보일 수 있습니다.

HTML5 Viewer로 보고서 서비스를 하는 경우 지원하지 않는 기능은 사용하지 않는 것이 좋습니다.

- **콤보 박스 객체**

폼필드 콤보 박스는 지원하지만, 일반 콤보 박스는 지원하지 않습니다.

일반 콤보 박스는 Designer의 삽입 > 객체 > 메뉴를 통해 제공되고 폼필드 콤보 박스는 삽입 > 폼필드 > 메뉴를 통해 제공됩니다.

일반 콤보 박스 객체는 HTML5 Viewer 화면에서 표시되지 않습니다.

- **폼필드 문자 속성**

디자이너에서 폼필드에 배분 및 양쪽 정렬, 텍스트 간격 속성 등을 적용하고 HTML5 Viewer에서 미리보기 혹은 데이터를 삽입할때는 속성이 적용되지 않습니다.

단 PDF 출력 및 export 시 텍스트 속성 등이 적용됩니다.

뷰어 기능 제약

- 음영 기능을 사용한 경우의 PDF 저장/인쇄 관련 제약

음영처리된 세로문자는 PDF로 출력이 되지 않습니다. 또한, **Ctrl+F8** 단축키로 세로 쓰기한 문자는 셀 밖으로 표시될 수 있습니다.

HTML5 Viewer				PDF변환			
	음영	장평	좌측정렬		음영	장평	좌측정렬
	음영,이탤릭,굵게	이탤릭&취소	이탤릭&음영		음영,이탤릭,굵게	이탤릭&취소	이탤릭&음영
세로쓰기	Shade	Shade	Shade	세로쓰기	Shade	Shade	Shade
	Shade	Shade	Shade		Shade	Shade	Shade
세로쓰기 (Ctrl+F8)	Shade	Shade	Shade	세로쓰기 (Ctrl+F8)	Shade	Shade	Shade
	Shade	Shade	Shade		Shade	Shade	Shade

- 제출 전 미리보기 제약

제출 전 미리보기에서 멀티라인 텍스트를 그릴 때는 항상 글자별로 자동 개행을 하기 때문에 뷰어와 다르게 보일 수 있습니다.

(제출 전 미리보기와 뷰어의 개행여부가 반드시 동일해야 할 경우 입력시 엔터를 통해 개행 문자를 삽입해야 합니다.)

- 보고서 검색 기능의 제약사항

[검색 제약사항](#)의 내용을 참조하십시오.

실행 환경에 의한 제약 사항

HTML5 Viewer는 브라우저에서 구동하는 뷰어이기 때문에 브라우저의 HTML5 스펙 구현 정도에 따라 브라우저마다 조금씩 다른 결과를 보일 수 있습니다.

따라서 일부 브라우저에서는 이 용에 제약을 받을 수 있고, 브라우저 설정에 따라 제약이 생길 수 있습니다. 다음은 브라우저에 및 모바일 환경 등에 의한 HTML5 Viewer의 제약 사항에 대한 설명입니다.

• XHR2

HTML5 Viewer는 서버와 XHR을 이용해 통신을 하는데, 데이터를 내려 받으면서 곧바로 화면에 보여주는 기능은 XHR2에 적용되어 있습니다.

IE9의 경우 XHR2가 적용되어 있지 않아 보고서를 전부 내려 받기 전에는 화면에 보고서가 보이지 않습니다.

그러므로 IE9에서 대용량의 보고서를 실행했을 때는 사용자가 다소 느리다고 느낄 수 있습니다.

IE9을 제외한 거의 모든 브라우저의 최신 버전들은 XHR2를 지원합니다.

• 텍스트 스타일

스타일	제약 환경	제약 환경에서의 실행 결과
장평	Internet Explorer, Chrome, Firefox, Safari 등의 브라우저가 해당 텍스트 속성을 지원하지 않습니다. (향후, 각 브라우저가 장평을 지원하게 되면 HTML5 Viewer에도 적용할 수 있습니다).	기본 글꼴 스타일로 표시됩니다
양쪽 정렬	브라우저의 'Justify' 속성과 동일합니다. Viewer에서는 셀 안에 문장을 라인 단위로 쪼개어 정렬하기 때문에 적용이 안됩니다. (브라우저 'Justify' 속성 참조)	왼쪽 정렬로 표시됩니다.

ü 브라우저 'Justify' 속성

Justify 속성은 브라우저에서 지원하는 **양쪽 정렬** 속성입니다. 한 셀 내에 여러 줄의 문장이 있을 때만 적용됩니다.

(한 줄 이하의 문장이 있으면 적용이 되지 않고 왼쪽 정렬로 표시됩니다).

안녕하세요. 엠투소프트입니다.
 테스트 환경 구축중입니다.
 HTML5 Viewer 양쪽 정렬 테스트
 입니다.

안녕하세요. 엠투소프트입니다.
 테스트 환경 구축중입니다.
 HTML5 Viewer 양쪽 정렬 테스트
 입니다.

안녕하세요. 엠투소프트입니다.
 테스트 환경 구축중입니다.
 HTML5 Viewer 양쪽 정렬 테스트
 입니다.

안녕하세요. 엠투소프트입니다.
 테스트 환경 구축중입니다.
 HTML5 Viewer 양쪽 정렬 테스트
 입니다.

EXE Viewer
 (상) 양쪽 정렬
 (하) 왼쪽 정렬

HTML5 Viewer
 (상) 양쪽 정렬
 (하) 왼쪽 정렬

ü 성능과 관련된 텍스트 스타일

배분 정렬 스타일 적용을 하면, 글자 한 개씩 셀에 맞게 계산하는 작업이 들어가기 때문에 글자 분량에 따라 속도가 저하될 수 있습니다.

● 도형의 점선

직선은 점선을 지원하지만 타원, 곡선, 자유 곡선 등의 도형에는 점선이 적용되지 않을 수 있습니다. 이는 HTML5 스펙에 점선을 그리는 API가 추가되었지만, 아직 브라우저에서 지원하지 않기 때문입니다.

브라우저에서 지원하지 않을 경우 점선이 실선으로 표시되며, 각 브라우저의 최신 버전에서 점차 지원되고 있습니다.



< safari browser(5.1.7)>



< Chrome browser(33.0.1750.149 m)>

● 웹 폰트

모바일 환경에서 서비스하려면 보고서를 디자인할 때 각 모바일 기기에서 사용 가능한 웹 폰트를 사용하는 것이 좋습니다.

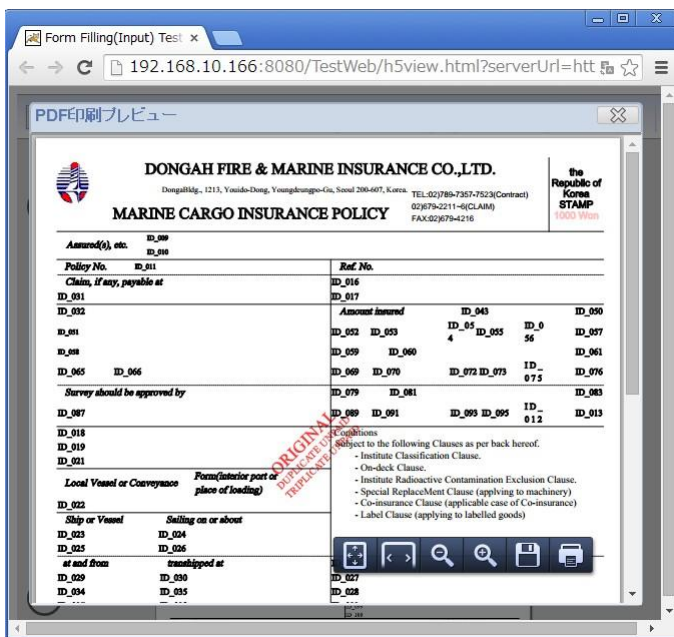
폰트를 ttf 확장자로도, woff 확장자로도 가지고 있어야 더 많은 브라우저에서 동일하게 보일 수 있습니다. 웹 폰트를 사용하지 않으면 보고서에 적용한 폰트로 보여주기 위해 별도로 폰트를 설치해야 합니다. 폰트가 설치되지 않으면, 보고서에 작성한 폰트와 다르게 표시될 수 있습니다.

웹 폰트에 관한 자세한 설명은 [웹 폰트 사용하기](#)의 설명을 참조하십시오.

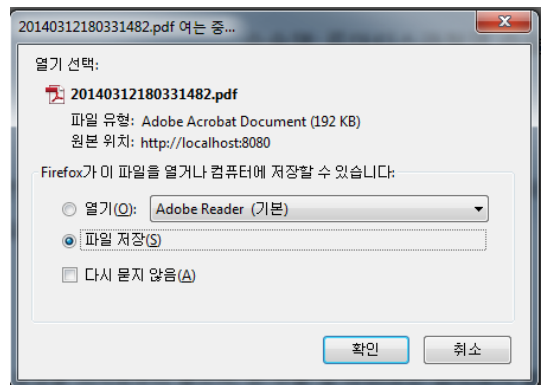
● PDF 인쇄

사용자 PC에서 PDF 인쇄를 요청하면 서버에서 작성된 PDF 문서가 사용자 PC로 다운로드됩니다. 이 때 브라우저의 설정에 의해 PDF 인쇄 미리 보기가 아닌 파일 저장 형식으로 실행될 수 있습니다.

PDF 인쇄 미리 보기 형식으로 실행하려면 각 브라우저의 Adobe Reader 플러그인을 활성화 상태로 설정해야 합니다.



< Adobe Reader 플러그인 방식 실행 >



< 파일 저장 방식 실행 >

● 클라이언트 사이트 PDF 기능의 브라우저 제약

저장 : IE9 이하에서는 지원하지 않습니다.

인쇄 : IE와 Edge에서는 지원하지 않습니다.

• Internet Explorer 7, 8 브라우저 (ERS 6.3.4.189 / Java Viewer 5.5.8.232 / HTML5 Viewer 6.3.4.158 부터 지원)

HTML5 Viewer에서는 JE7/IE8 브라우저를 지원하기 위해 Mobile Server의 이미지 기반 보고서를 사용합니다.

다음에 나열된 기능만 지원하며 API, 암호화 등 기타 기능을 사용할 수 없습니다.

기능	상세 기능
보고서 뷰잉	리포팅 중지
페이지 이동	처음 페이지 마지막 페이지 이전 페이지 이후 페이지 페이지 이동
페이지 확대/축소	확대 축소
Export	PDF DOC XLS PPT HWP
Print	PDF Print

• Cross-Domain 환경 지원 스펙

Cross-Domain 환경이란 보고서를 요청한 페이지와 ReportingServer의 도메인이 다를 경우를 의미합니다.

브라우저	상세 설명
Internet Explorer 7 이하	사용 불가 (브라우저에서 지원해주는 기술이 없습니다).
Internet Explorer 8	표준 방식에서는 지원하지 않아, 우회 방안을 적용했습니다. 대신 POST 방식을 지원하지 못하는 제약 사항이 발생합니다. GET 방식만을 지원하기 때문에 긴 파라미터를 사용할 수 없습니다. (HTML5 Viewer 6.4.0.217 버전부터 POST 방식을 지원합니다.)
Internet Explorer 9	사용자가 아래 설정을 사용함으로 설정해야 사용 가능합니다. 인터넷 옵션 보안 사용자 지정 수준 도메인간의 데이터 원본 액세스
Internet Explorer 10 이상 Chrome Safari Opera Firefox	표준 방식으로 지원합니다.

• 구간 암호화 사용 제한

openFile 옵션 중 downloadProtocolFile : true로 사용 시 아래와 같이 설정해서 사용하는 구간 암호화를 사용할 수 없습니다.

또한, 첨부파일을 포함한 보고서는 구간 암호화를 사용할 수 없습니다.

HTML5 Viewer의 option `{enableSecure: true}`

ReportingServer의 server.properties 값에 `secure.enable=true`

- 드릴다운 및 라이트 제약 사항

드릴다운이나 라이트 시, 객체 옆에 표나 다른 객체가 있으면 동작하지 않습니다.

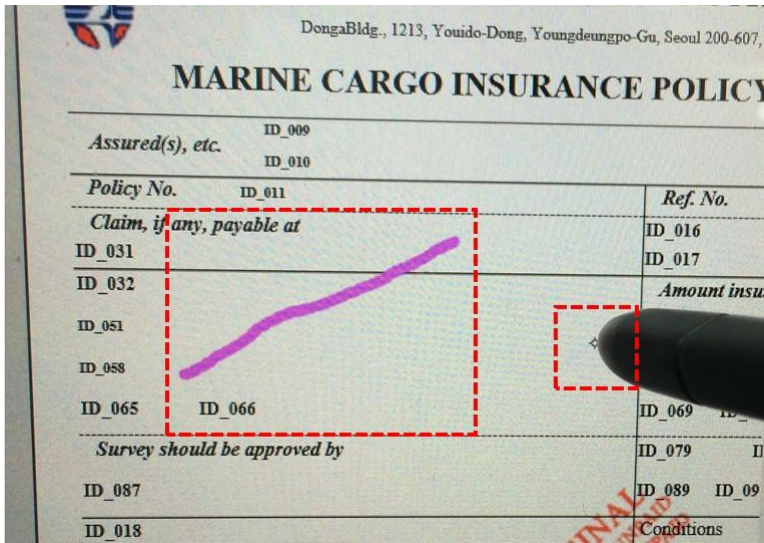
기타 환경에 의한 제약 사항

일부 모바일 기기나 스마트 PC에서 발생하는 제약사항에 대하여 설명합니다.

- 투 핀치 줌 이후 노트 표시

투 핀치 줌을 통해 보고서를 확대 축소 한 후 노트 작성 시 잘못된 위치에 그려질 수 있습니다.

특정 장비에 한해 발생하며, 이 경우 투 핀치 줌 사용을 제한하는 스크립트 코드를 이용하여 오류를 방지할 수 있습니다



```
<div id="crownix-viewer" style="position:absolute; width:100%; height:100%;-ms-touch-action:none"></div>
```


IE8 환경에서 지원하지 않는 API

IE8 환경에서 지원하지 않는 API 목록입니다.

- **zoom(ratio)**
- **bind(string eventName, function eventHandler)**
- **useWebFont(boolean use)**
- **useAdjustLetterSpace(use)**
- **setStateLessMode(isStateless)**
- **submit(url, datasource, servicecode, target)**

IE8 환경은 이미지방식의 HTML Viewer이기 때문에 폼필드 기능을 사용하지 못합니다.

트리 그리드 보고서 제약 사항

- 동작 조건

- 반복부 표 -표 속성 -데이터 그리드 -표를 그리드 객체로 사용 체크

- 반복부 표 -모든 요약부의 요약 조건 -반복부 앞에 삽입 체크

- 데이터 그룹핑(Depth 설정) 은 요약함수를 사용해야 합니다.
(반복부에 대한 요약 (sum, avg, count 등)을 설정해야 합니다.)

- 사용 불가 기능 목록

- 크로스 탭 ,다단 지정 ,객체 변수 ,등근 표

- 셀 병합 ,칸 합치기 ,한번만 쓰기

- 반복부 /요약부 선 속성

- 런타임 동작 옵션 제약

- 글꼴 크기 조정/장평 조정 옵션은 동작하지 않습니다.

- 런타임 요약 함수는 동작하지 않습니다.

- 예) `changeattr()`, `changerowattr()`, `changetextattr()`, `exec()`, `deleterow()`, `deletecolumn()`, `mergecell()`, `mergecellex()` 등

- 복잡한 모양의 표를 트리그리드로 사용할 경우 Designer에서 만든 표와 다를 수 있습니다.

- 속성 별 제약 목록

- 화살표 ,음영 ,장평 ,위 첨자 ,아래 첨자 ,여백은 지원하지 않습니다.

- 폰트 제약 : 셀 안에서 글자마다 글꼴 속성을 다르게 할 수 없습니다 Designer의 폰트 크기와 위치가 달라질 수 있습니다.참고로 속성의 최소 지정 단위는 셀 *입니다 .

- 그 외 지원하지 않는 항목

텍스트 상자 속성
?
×

텍스트 상자 속성
텍스트 상자 종류

텍스트 방향

가로

세로

상하 정렬

상단 정렬

중앙 정렬

하단 정렬

좌우 정렬

왼쪽 정렬

가운데 정렬

오른쪽 정렬

양쪽 정렬

배분 정렬

여백

상 pt 하 pt

좌 pt 우 pt

문자 간격 inch (1/1000) 들여쓰기 pt

줄 간격 pt

메모 객체 인쇄 제외

쓰기 보호 HTML 소스 변환

셀 크기 고정

글꼴 크기 조정

장평 조정

확인
취소
적용(A)

틀고정 보고서 제약사항

- 사용 불가 보고서 기능 목록
 - ü 폼필드 기능
 - ü 드릴 오버 기능
 - ü 레이어 기능

- 사용 불가 뷰어 기능 목록
 - ü 노트 기능
 - ü 하이퍼링크의 하버(Hover) 효과