



시작 가이드

How to Start

Contents

시작하기 전에	03

SpreadJS에 프로젝트 추가	04

SpreadJS에 데이터 바인딩	06

SpreadJS에 스타일 설정	08

Excel 가져오기 / 내보내기	10

SpreadJS 라이브러리 목록	13

시작하기 전에

SpreadJS는 스프레드시트 컴포넌트입니다. 자주 사용되는 기능으로는 수식 엔진, 정렬, 필터링, 입력 제어, 스파크라인 및 Excel 입/출력이 있습니다. 이 문서에서는 SpreadJS를 프로젝트에 추가하는 방법과 데이터 바인딩, 스타일 설정에 대해 설명합니다.

해당 자료는 SpreadJS를 처음 사용하시는 분들을 위한 만들어진 시작 가이드로, 조금 더 자세한 정보와 소식을 보기 위해서는 아래 링크를 참고해 주시기 바랍니다.

[제품 정보 확인]

- SpreadJS : <https://www.grapecity.co.kr/spreadjs>
- SpreadJS 다운로드 : <https://www.grapecity.co.kr/download/spreadjs>
- SpreadJS 디자이너 소스코드 : <https://spreadjs-designer-sc.web.app/index/index.html>

[개발 가이드]

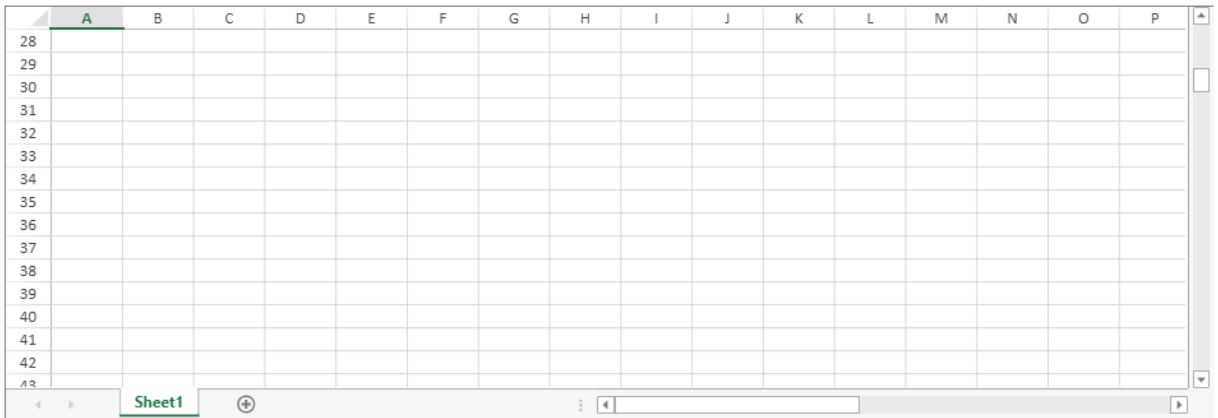
- SpreadJS 튜토리얼 센터 : <https://www.grapecity.co.kr/spreadjs/demo>
- SpreadJS 기술 커뮤니티 : <https://cafe.naver.com/grapecity>
- SpreadJS 개발자 문서 : <https://www.grapecity.co.kr/support/resources#SpreadJS>

SpreadJS 라이브러리

- SpreadJS 필수 라이브러리 파일 경로 (`gc.spread.sheets.all.xx.x.x.min.js`)
>> `SpreadJS.Release.xx.x.x\SpreadJS\scripts`
- SpreadJS CSS 파일 경로
>> `SpreadJS.Release.xx.x.x\SpreadJS\css`
- SpreadJS의 ExcelIO 라이브러리 파일 경로 (`gc.spread.excelio.xx.x.x.min.js`)
>> `SpreadJS.Release.xx.x.x\SpreadJS\scripts\winterop`
- SpreadJS 언어 리소스 파일 경로
>> `SpreadJS.Release.xx.x.x\SpreadJS\scripts\resources`
- `gc.spread.sheets.all.xx.x.x.min.js`는 필수적으로 추가되어 있어야 하며 **바코드, 차트, 엑셀 입출력** 등과 같은 기능들에 대해서는 각각에 해당하는 라이브러리를 별도로 추가해 주어야 합니다.

SpreadJS를 프로젝트에 추가

해당 섹션에서는 아래 결과 이미지와 같이 SpreadJS를 웹 상에 초기화하는 방법을 구현합니다. 이를 위해, 먼저 SpreadJS 라이브러리들을 추가하는 방법에 대해서 설명합니다.



Step 1.

Excel UI 화면 구현을 위한 CSS 파일인 **gc.spread.sheets.xx.x.x.css**와, Excel의 기능을 위한 필수 JS 라이브러리, **gc.spread.sheets.all.xx.x.x.min.js**를 선언합니다. 또한, 한국어 지원이 필요한 경우, **gc.spread.sheets.resources.ko.xx.x.x.min.js** 언어 리소스도 함께 선언 합니다. (기본 영어)

```
1 <link rel="styleSheet" href="gc.spread.sheets.xx.x.x.css" />
2 <script src="gc.spread.sheets.all.xx.x.x.min.js" type="text/javascript"> </script>
3 <script src="gc.spread.sheets.resources.ko.xx.x.x.min.js" type="text/javascript"> </script>
```

Step 2.

DOM 요소를 페이지 Body에 컨테이너로 포함합니다.

```
1 <div id="ss" style="width:100%; height:360px;border: 1px solid gray;"> </div>
```

Step 3.

onload 시점에 JavaScript를 통하여, 아래와 같이 div 태그 상에 시트의 개수가 1개인 SpreadJS 시트를 초기화합니다.

```

1 window.onload = function () {
2     var spread = new GC.Spread.Sheets.Workbook(document.getElementById('ss'), { sheetCount:
3     1 });
4 };

```

전체 코드

```

1 <!doctype html>
2 <html style="height:100%;font-size:14px;">
3
4 <head>
5     <meta name="spreadjs culture" content="ko-kr" />
6     <meta charset="utf-8" />
7     <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8
9     <link rel="styleSheet" href="css/spread/gc.spread.sheets.excel2013white.12.2.5.css" />
10    <script src="js/spread/gc.spread.sheets.all.12.2.5.min.js" type="text/javascript"></script>
11    <script src="js/spread/resources/ko/gc.spread.sheets.resources.ko.12.2.5.min.js"
12        type="text/javascript"></script>
13
14 </head>
15 <body style='font-size:14px;'>
16
17     <div id="ss" style="width:100%; height:360px;border: 1px solid gray;">
18     </div>
19
20     <script type="text/javascript">
21         window.onload = function() {
22             var spread = new GC.Spread.Sheets.Workbook(
23                 document.getElementById('ss'), { sheetCount: 1 });
24         };
25     </script>
26
27 </body>
28 </html>

```

SpreadJS에 데이터 바인딩

이번 장에서는 SpreadJS에 Cell 바인딩 메소드를 사용하여 특정 셀 영역에 데이터를 표시하는 방법을 설명합니다.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1																
2																
3			Wang feng													
4			25													
5			Male													
6			710075													
7																
8																
9																
10																
11																
12																
13																
14																
15																
16																

Step 1.

먼저, `getSheet` 함수를 사용하여, 활성화하고자 하는 Sheet를 가져옵니다.

“`new GC.Spread.Sheets.Bindings.CellBindingSource(person);`”를 사용하여 “`person`” 이라는 json 형식의 데이터를 셀 바인딩을 위한 데이터소스로 설정합니다.

그 다음 `setBindPath` 메서드를 사용하여 현재 시트에서 데이터를 보여줄 특정 셀 영역을 설정합니다.

마지막으로, `setDataSource` 메서드를 사용하여 시트 상에 데이터소스를 바인딩 합니다.

```

1 var sheet = spread.getSheet(0);
2 var person = { name: 'Wang feng', age: 25, gender: 'Male', address: { postcode: '710075' } };
3 var source = new GC.Spread.Sheets.Bindings.CellBindingSource(person);
4 sheet.setBindingPath(2, 2, 'name');
5 sheet.setBindingPath(3, 2, 'age');
6 sheet.setBindingPath(4, 2, 'gender');
7 sheet.setBindingPath(5, 2, 'address.postcode');
8 sheet.setDataSource(source);

```

전체 코드

```
1 <!doctype html>
2 <html style="height:100%;font-size:14px;">
3
4 <head>
5   <meta name="spreadjs culture" content="ko-kr" />
6   <meta charset="utf-8" />
7   <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8   <link rel="styleSheet" href="css/spread/gc.spread.sheets.excel2013white.12.2.5.css" />
9   <script src="js/spread/gc.spread.sheets.all.12.2.5.min.js" type="text/javascript"> </script>
10  <script src="js/spread/resources/ko/gc.spread.sheets.resources.ko.12.2.5.min.js"
    type="text/javascript">
11 </script>
12 </head>
13
14 <body style='font-size:14px;'>
15
16   <div id="ss" style="width:100%; height:360px;border: 1px solid gray;"> </div>
17
18   <script type="text/javascript">
19     window.onload = function() {
20       var spread = new GC.Spread.Sheets.Workbook(document.getElementById('ss'), {
21         sheetCount: 1
22       });
23
24       var sheet = spread.getSheet(0);
25
26       var person = {
27         name: 'Wang feng',
28         age: 25,
29         gender: 'Male',
30         address: {
31           postcode: '710075'
32         }
33       };
34
35       var source = new GC.Spread.Sheets.Bindings.CellBindingSource(person);
36       sheet.setBindingPath(2, 2, 'name');
37       sheet.setBindingPath(3, 2, 'age');
38       sheet.setBindingPath(4, 2, 'gender');
39       sheet.setBindingPath(5, 2, 'address.postcode');
40       sheet.setDataSource(source);
41     };
42   </script>
43
44 </body>
45
46 </html>
```

SpreadJS에 스타일 설정

이번 장에서는 SpreadJS의 Sheet를 더 시각적으로 만들고 표준화하기 위해 시트의 스타일을 설정하는 방법을 설명합니다.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1														
2		Person Card												
3		Name: Wang feng												
4		Age: 25												
5		Gender: Male												
6		Address: Postcode: 710075												
7														
8														
9														
10														
11														
12														
13														
14														
15														
16														

Step 1.

이전 섹션에서 데이터가 바인딩 된 Sheet를 사용자가 더욱 쉽게 데이터를 볼 수 있도록 각 데이터 항목의 스타일을 설정합니다.

SpreadJS에 텍스트를 추가하여 각 데이터에 대한 컬럼 헤더를 생성하고, "addSpan" 메서드를 사용하여 셀을 병합하여 하나의 제목 셀을 생성합니다.

지정된 열의 너비, 배경색(backColor), 글자색(forecolor), 테두리(border), 정렬(horizontalAlign) 등과 다양한 스타일을 속성을 추가할 수도 있습니다.

```

1 sheet.getCell(2,1).text("Name");
2 sheet.getCell(3,1).text("Age");
3 sheet.getCell(4,1).text("Gender");
4 sheet.getCell(5,1).text(" Address.Postcode");
5 sheet.addSpan(1, 1, 1, 2);
6 sheet.getRange(1, 1, 1, 2).text("Person Card")
7 sheet.setColumnWidth(1, 120);
8 sheet.setColumnWidth(2, 120);
9 sheet.getRange(1, 1, 1, 2).backColor("rgb(20, 140, 1218)")
10 sheet.getRange(1, 1, 1, 2).foreColor("rgb(255, 255, 255)")
11 sheet.getRange(2, 1, 4, 1).backColor("rgb(169, 238, 243)")
12 sheet.getRange(2, 1, 4, 1).foreColor("rgb(255, 255, 255)")
13 sheet.getRange(2, 2, 4, 1).backColor("rgb(247, 197, 113)")
14 sheet.getRange(1, 1, 5, 2).setBorder(new GC.Spread.Sheets.LineBorder("Black",
GC.Spread.Sheets.LineStyle.thin), {
15     all: true
16 });
17 sheet.getRange(2, 1, 4, 2).setBorder(new GC.Spread.Sheets.LineBorder("Black",
GC.Spread.Sheets.LineStyle.dotted), {
18     inside: true
19 });
20 sheet.getRange(1, 1, 1, 2).hAlign(GC.Spread.Sheets.HorizontalAlign.center);

```


전체 코드

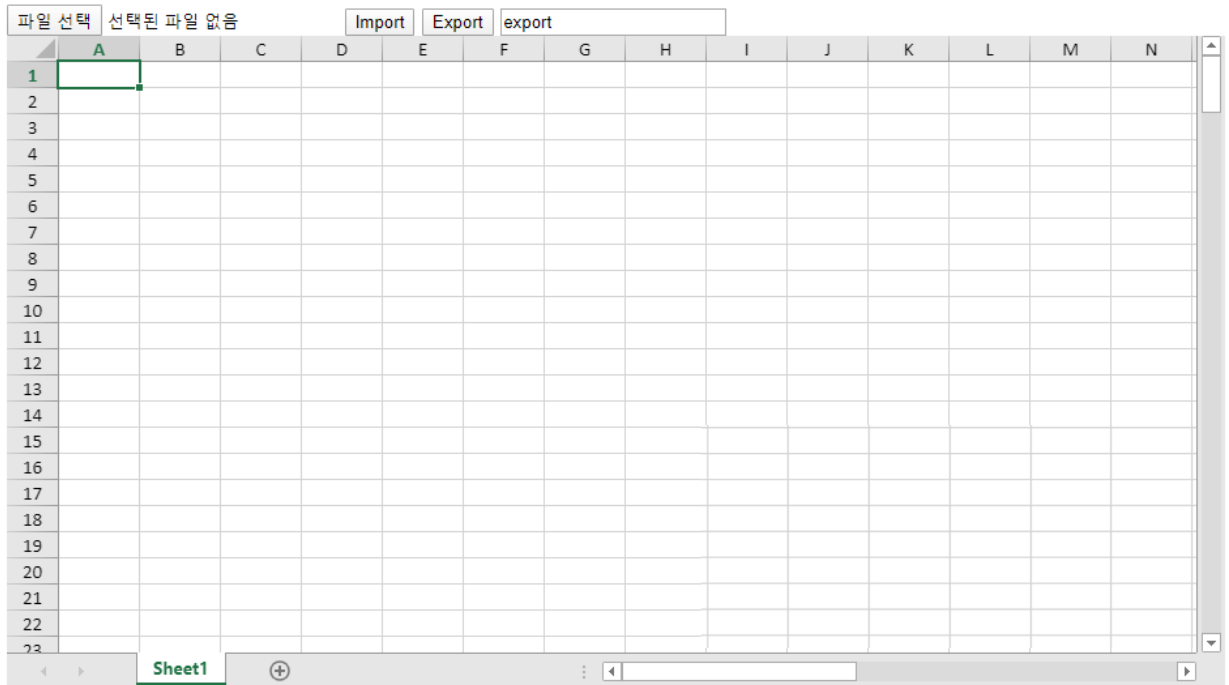
```

1 <!doctype html>
2 <html style="height:100%;font-size:14px;">
3
4 <head>
5 <meta name="spreadjs culture" content="ko-kr" />
6 <meta charset="utf-8" />
7 <meta name="viewport" content="width=device-width, initial-scale=1.0" />
8 <link rel="styleSheet" href="css/spread/gc.spread.sheets.excel2013white.12.2.5.css" />
9 <script src="js/spread/gc.spread.sheets.all.12.2.5.min.js" type="text/javascript"> </script>
10 <script src="js/spread/resources/ko/gc.spread.sheets.resources.ko.12.2.5.min.js" type="text/javascript">
11 </script>
12 </head>
13
14 <body style='font-size:14px;'>
15 <div id="ss" style="width:100%; height:360px;border: 1px solid gray;"> </div>
16 <script type="text/javascript">
17     window.onload = function() {
18         var spread = new GC.Spread.Sheets.Workbook(document.getElementById('ss'), {
19             sheetCount: 1
20         });
21         var sheet = spread.getSheet(0);
22         var person = {
23             name: 'Wang feng',
24             age: 25,
25             gender: 'Male',
26             address: {
27                 postcode: '710075'
28             }
29         };
30         var source = new GC.Spread.Sheets.Bindings.CellBindingSource(person);
31         sheet.setBindingPath(2, 2, 'name');
32         sheet.setBindingPath(3, 2, 'age');
33         sheet.setBindingPath(4, 2, 'gender');
34         sheet.setBindingPath(5, 2, 'address.postcode');
35         sheet.setDataSource(source);
36         sheet.getCell(2, 1).text("Name");
37         sheet.getCell(3, 1).text("Age");
38         sheet.getCell(4, 1).text("Gender");
39         sheet.getCell(5, 1).text("Address.Postcode");
40         sheet.addSpan(1, 1, 1, 2);
41         sheet.getRange(1, 1, 1, 2).text("Person Card")
42         sheet.setColumnWidth(1, 120);
43         sheet.setColumnWidth(2, 120);
44         sheet.getRange(1, 1, 1, 2).BackColor("rgb(20, 140, 1218)")
45         sheet.getRange(1, 1, 1, 2).ForeColor("rgb(255, 255, 255)")
46         sheet.getRange(2, 1, 4, 1).BackColor("rgb(169, 238, 243)")
47         sheet.getRange(2, 1, 4, 1).ForeColor("rgb(255, 255, 255)")
48         sheet.getRange(2, 2, 4, 1).BackColor("rgb(247, 197, 113)")
49         sheet.getRange(1, 1, 5, 2).setBorder(new GC.Spread.Sheets.LineBorder("Black",
50             GC.Spread.Sheets.LineStyle.thin, {
51                 all: true
52             }));
53         sheet.getRange(2, 1, 4, 2).setBorder(new GC.Spread.Sheets.LineBorder("Black",
54             GC.Spread.Sheets.LineStyle.dotted), {
55                 inside: true
56             }));
57         sheet.getRange(1, 1, 1, 2).hAlign(GC.Spread.Sheets.HorizontalAlign.center);
58     };
59 </script>
60 </body>
61 </html>

```

Excel 가져오기 / 내보내기

이번 장에서는 SpreadJS의 ExcellIO 기능을 통해 쉽고 빠르게 Excel 파일(.xlsx)을 가져오거나 내보내는 기능을 구현 하는 방법을 설명합니다.



Step 1.

SpreadJS에서 ExcellIO 기능을 사용하려면 관련 라이브러리를 추가해 주어야 합니다.

> 파일 경로:

SpreadJS.Release.xx.x\WSpreadJS\Scripts\Winterop\gc.spread.excelio.x.xx.xxxxx.x.min.js

```

1 <head>
2   ...
3   <script src='../spreadjs/gc.spread.sheets.all.xx.xxxxx.x.min.js' type='text/javascript'> </script>
4   <script src='../spreadjs/plugins/gc.spread.excelio.x.xx.xxxxx.x.min.js'
5   type='text/javascript'> </script>
6 </head>

```

Step 2.

SpreadJS와 ExcellIO 인스턴스를 초기화합니다.

```

1 var spread = new GC.Spread.Sheets.Workbook(document.getElementById('ss'));
2 var excelio = new GC.Spread.Excel.IO();

```

Step 3.

ExcelIO의 `open()` 또는 `save()` 함수를 사용하여 Excel 파일(.xlsx)을 SpreadJS로 가져오거나 SpreadJS를 Excel 파일로 내보낼 수 있습니다.

```
1 //import excel file to Spread.Sheets json
2 excelio.open(excelFile, function (json) {
3     var workbookObj = json;
4     workbook.fromJSON(workbookObj);
5 }, function (e) {
6     // process error
7     console.log(e);
8 });
9 //export Spread.Sheets json to excel file
10 excelio.save(json, function (blob) {
11     //do whatever you want with blob
12     //such as you can save it
13 }, function (e) {
14     //process error
15     console.log(e);
16 });
```

전체 코드

```

1 <!DOCTYPE html>
2 <html>
3 <head>
4   <script src="https://cdnjs.cloudflare.com/ajax/libs/FileSaver.js/2014-11-29/FileSaver.min.js" > </script>
5   <link href="css/spread/gc.spread.sheets.excel2016colorful.12.2.5.css" rel="stylesheet"/>
6   <script src="js/spread/gc.spread.sheets.all.12.2.5.min.js" type="application/javascript" > </script>
7   <script src="js/spread/gc.spread.excelio.12.2.5.min.js" > </script>
8 </head>
9 <body>
10  <!-- 화면 구현 -->
11  <div>
12    <input type="file" name="files[]" id="fileDemo" accept=".xlsx,.csv"/>
13    <input type="button" id="loadExcel" value="Import" onclick="ImportFile()"/>
14    <input type="button" class="btn btn-default" id="saveExcel" value="Export" onclick="ExportFile()"/>
15    <input type="text" id="exportFileName" placeholder="Export file name" class="form-control"
16    value="export"/>
17    <div id="ss" style="width:100%;height:500px" > </div>
18  </div>
19 </body>
20 <script>
21   var workbook, excelIO;
22   window.onload = function () {
23     workbook = new GC.Spread.Sheets.Workbook(document.getElementById("ss"));
24     excelIO = new GC.Spread.Excel.IO();
25     var sheet = workbook.getSheet(0);
26   }
27  <!-- Excel 불러오기 -->
28  function ImportFile() {
29
30     workbook.suspendPaint();
31     workbook.suspendCalcService();
32     workbook.suspendEvent();
33     var excelFile = document.getElementById("fileDemo").files[0];
34     excelIO.open(excelFile, function (json) {
35       var workbookObj = json;
36       workbook.fromJSON(workbookObj, {
37         doNotRecalculateAfterLoad: true
38       });
39     }, function (e) {
40       console.log(e);
41     });
42     workbook.resumeEvent();
43     workbook.resumeCalcService();
44     workbook.resumePaint();
45
46   }
47  <!-- Excel 저장 -->
48  function ExportFile() {
49
50     var fileName = document.getElementById("exportFileName").value;
51     if (fileName.substr(-5, 5) !== '.xlsx') {
52       fileName += '.xlsx';
53     }
54     var json = workbook.toJSON();
55     excelIO.save(json, function (blob) {
56       saveAs(blob, fileName);
57     }, function (e) {
58       console.log(e);
59     });
60
61   }
62 </script>
63 </html>

```

SpreadJS 라이브러리 목록

Features	Required Libraries
Core	gc.spread.common.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js
Calc Engine	gc.spread.common.*.*.*.min.js gc.spread.calcengine.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js
Basic Functions	gc.spread.common.*.*.*.min.js gc.spread.calcengine.*.*.*.min.js gc.spread.calcengine.basicfunctions.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js
Advanced Functions	gc.spread.common.*.*.*.min.js gc.spread.calcengine.*.*.*.min.js gc.spread.calcengine.advancedfunctions.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js
Data Binding	gc.spread.common.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.bindings.*.*.*.min.js
Drag Fill	gc.spread.common.*.*.*.min.js gc.spread.calcengine.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.fill.*.*.*.min.js
Filter	gc.spread.common.*.*.*.min.js gc.spread.calcengine.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.conditionalformatting.*.*.*.min.js gc.spread.sheets.filter.*.*.*.min.js
Formula Textbox	gc.spread.common.*.*.*.min.js gc.spread.calcengine.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.formulatextbox.*.*.*.min.js
Floating Object	gc.spread.common.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.floatingobjects.*.*.*.min.js
Outline	gc.spread.common.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.outlines.*.*.*.min.js
Touch	gc.spread.common.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.touch.*.*.*.min.js
Print	gc.spread.common.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.print.*.*.*.min.js

SpreadJS 라이브러리 목록

Features	Required Libraries
Cell Type	gc.spread.common.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.celltypes.*.*.*.min.js
Comment	gc.spread.common.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.comments.*.*.*.min.js
Conditional Formatting	gc.spread.common.*.*.*.min.js gc.spread.calcengine.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.conditionalformatting.*.*.*.min.js
Data Validation	gc.spread.common.*.*.*.min.js gc.spread.calcengine.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.conditionalformatting.*.*.*.min.js gc.spread.sheets.datavalidation.*.*.*.min.js
Search	gc.spread.common.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.search.*.*.*.min.js
Sparkline	gc.spread.common.*.*.*.min.js gc.spread.calcengine.*.*.*.min.js gc.spread.calcengine.basicfunctions.*.*.*.min.js gc.spread.calcengine.advancedfunctions.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.sparklines.*.*.*.min.js
Table	gc.spread.common.*.*.*.min.js gc.spread.calcengine.*.*.*.min.js (선택, for formulas and filters) gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.bindings.*.*.*.min.js (선택, for data-binding) gc.spread.sheets.conditionalformatting.*.*.*.min.js (선택, for filtering) gc.spread.sheets.filter.*.*.*.min.js (선택, for filtering) gc.spread.sheets.tables.*.*.*.min.js
Chart	gc.spread.common.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.bindings.*.*.*.min.js (for data-binding) gc.spread.sheets.charts.*.*.*.min.js
Drag Merge	gc.spread.common.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.dragmerge.*.*.*.min.js

SpreadJS 라이브러리 목록

Features	Required Libraries
Outline Column	gc.spread.common.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.outlinecolumn.*.*.*.min.js
Context Menu	gc.spread.common.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.contextmenu.*.*.*.min.js gc.spread.sheets.bindings.*.*.*.min.js (for data-binding)
PDF	gc.spread.common.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.pdf.*.*.*.min.js
Slicer Component	gc.spread.common.*.*.*.min.js
Slicer	gc.spread.common.*.*.*.min.js gc.spread.calcengine.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.conditionalformatting.*.*.*.min.js gc.spread.sheets.filter.*.*.*.min.js gc.spread.sheets.tables.*.*.*.min.js gc.spread.sheets.floatingobjects.*.*.*.min.js gc.spread.sheets.slicers.*.*.*.min.js
Barcode	gc.spread.common.*.*.*.min.js gc.spread.calcengine.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.sparklines.*.*.*.min.js gc.spread.sheets.barcode.*.*.*.min.js
Shapes	gc.spread.common.*.*.*.min.js gc.spread.calcengine.*.*.*.min.js gc.spread.sheets.core.*.*.*.min.js gc.spread.sheets.floatingobjects.*.*.*.min.js gc.spread.sheets.shapes.*.*.*.min.js

<https://help.grapacity.com/spread/SpreadSheets12/webframe.html#modules.html>