

QCELL 사용자 매뉴얼

기초부터 활용까지

요약

웹 그리드인 큐셀을 정확하게 이해하고 확실하게 사용할 수 있도록 만든 문서로
이 문서로 전반적인 이해 후 툴킷, API 문서와 함께 활용하여 개발하십시오.

(주)라잇텍 기술지원팀

service@righttech.co.kr

목차

1 장. QCELL 이해하기	11
웹 그리드란	11
시스템 개발과 웹 그리드	11
QCELL 소개	12
기반 기술이 다른 그리드와 비교	13
2 장. 구성요소	17
1. cell	17
2. row 와 column(col)	17
3. frozenrow, frozenbottomrow 와 frozencol	18
4. scrollbar	19
3 장. 준비와 설치하기	20
1) QCELL 라이브러리	20
2) QCELL 설치	20
2.1 시스템 요구 사항	20
2.2 QCELL 제품 구성.....	21
1) 라이브러리	21
2) 사용자 매뉴얼	21
3) API 문서.....	21
5) 라이선스 증서	21
2.3 QCELL 패키지 복사.....	22
2.4 QCELL 라이브러리 설치 확인하기	23
4 장. 생성하기	24
1. QCELL 생성하기	24
1.1. 부모 컨테이너 선언.....	25
1.2. QCELL 속성 설정.....	25
2. QCELL Property	27
2.1 QCELL Property	27

2.2 QCELL Columns Property.....	34
5 장. 기능 이해하기	38
1. QCELL 생성/제거하기	39
1) create	39
2) hasInstance.....	39
3) getInstance	40
4) getInstanceIDs.....	40
5) deleteInstance	40
6) deleteAllInstances.....	41
2. 초기화	41
1) rebuild.....	41
2) refresh.....	43
3) resetData	43
4) initGrid.....	43
3. Property & Index	44
1) getID.....	44
2) getIdx.....	44
3) getRows	46
4) getCols	47
5) getRowHeight	48
6) getColWidth(s)	48
4. 데이터	49
4.1. 데이터 얻기(반환), 갱신하기	49
1) getData.....	49
2) getAllData	49
3) getAllDataWithFormat.....	50
4) setData	50
5) getRowData.....	52
6) setRowData	52
7) setRowDataEx.....	53
8) getFocusRowData.....	53
9) findRow	53

10) findRowEx.....	54
11) swapData.....	55
12) sort.....	55
13) getColData.....	56
14) setColData.....	57
15) getCellData.....	58
16) setCellData.....	58
17) getCellLabel.....	59
18) setCellLabel	60
19) getCellDataEx	60
20) setCellDataEx.....	61
21) getInitCellDataEx.....	61
4.2. 추가/수정/삭제.....	63
1) addRow	63
2) addRows	64
3) insertRow.....	65
4) insertRows.....	65
5) removeRow.....	66
6) removeRows.....	67
7) deleteRow	68
8) deleteRows	68
9) insertStaticRow	70
10) deleteStaticRow.....	70
11) isStaticRow.....	71
12) getState.....	71
13) clearStates.....	72
14) getStates.....	72
15) getStatesEx	73
16) setRowState.....	74
17) getRowState.....	75
18) removeRowState	76
19) clearRowStates.....	76
20) getFocusRowState.....	77

21) getFocusRowData.....	77
22) getSelectRowState.....	77
23) getStatesBySep.....	78
24) getAllStatesBySep.....	79
4.3. 기타.....	79
1) requestData.....	79
2) checkRequired.....	80
3) showProgress.....	81
4) hideProgress.....	81
5) editCell.....	82
6) closeEditMode.....	82
7) divisionSubmit.....	83
8) getUpdateInfo.....	84
4.4. 출력/숨김.....	84
1) setCellInvisible.....	84
4.5. 초기화.....	86
1) resetData.....	86
4.6. 정렬.....	86
1) sort.....	86
5. Selection.....	87
5.1. 선택 방법 설정.....	87
5.2. 로우/컬럼 인덱스 반환.....	89
1) getSelectedRow.....	89
2) getSelectedRows.....	89
3) getSelectedRowData.....	90
4) getSelectedCol.....	91
5) getSelectedCols.....	91
6. Scroll.....	92
1) scrollTop.....	92
2) scrollLeft.....	93
3) isScroll.....	94
7. Style.....	95

7.1. 높이 설정&반환	95
1) rowHeight	95
2) getRowHeight	95
7.2. 스타일 지정.....	96
1) setColStyle	96
2) removeColStyle.....	97
3) setRowStyle	97
4) removeRowStyle.....	98
5) getCellStyle.....	99
6) setCellStyle	99
7) setCellStyles	100
8) removeCellStyle.....	101
9) removeCellStyles.....	101
10) clearCellStyles.....	102
11) clearDynamicStyles	102
8. Column.....	103
8.1. 넓이 설정	103
8.2. 지원 유형	104
8.3. 컬럼 고정	105
8.4. 체크 박스	105
1) wholeselect.....	106
2) setHeaderChecked	106
3) getHeaderCheckedState	107
8.5. 컬럼 추가.....	107
1) insertColumn	107
8.6. 컬럼 삭제	108
1) removeColumn	108
8.7. 컬럼 초기화.....	110
1) resetColumns.....	110
8.8. 컬럼 합계	111
1) getColSum.....	111
8.9. 셀렉트 메뉴 컬럼과 label	112
1) setSelectmenuValue	112

2) getSelectmenuLabel	113
8.10. 필수 컬럼 정보 반환	113
1) checkRequired	113
8.11. 특정 위치 이동	114
1) moveCol	114
8.12. 컬럼 정보	115
1) getColumnInfo	115
2) getCurrentColumnsInfo	116
9. Row	117
9.1. 틀 고정	117
1) froszenrows	117
2) frozenbottomrows	117
3) setFrozenRows	118
4) getFrozenRows	119
5) setFrozenCols	120
6) getFrozenCols	120
7) setFrozenBottomRows	121
8) getFrozenBottomRows	122
9.2. 행 구분	122
1) highlightrepeat	122
2) getHighlightRepeat	123
3) setHighlightRepeat	124
10. Col	124
10.1. key 속성	124
1) getColOfKey	124
2) getKeyOfCol	125
10.2. 컬럼 ID/Type/Index 반환	125
1) getColID	125
2) getColOfID	126
3) getColType	126
4) getColAttribute	127
10.3. 넓이 설정 및 반환	127

1) getColWidth.....	127
2) setColWidth.....	128
11. 활성화/비활성화.....	129
11.1. QCELL.....	129
1) setDisable.....	129
2) getDisable.....	129
11.2. Cell.....	129
1) setCellDisable.....	129
2) setCellsDisable.....	130
11.3. Row.....	130
1) setRowDisable.....	130
11.4. Col.....	131
1) setColDisable.....	131
12. Pagination.....	132
1) setPageIndex.....	133
2) getPageIndex.....	133
3) setPageUnit.....	134
4) getPageUnit.....	134
5) setPaginationParam.....	134
6) getPaginationParam.....	135
7) setPaginationDatacount.....	135
8) getPaginationDatacount.....	136
9) setPaginationUrl.....	136
10) getPaginationUrl.....	136
11) setPaginationLoadonce.....	137
12) getPaginationLoadonce.....	137
13) setPaginationInitSort.....	137
13. 선행 컬럼(rowheader).....	137
13.1. 넘버링(Numbering) 방식.....	138
13.2. checkbox 방식.....	138
1) getRowheaderChecked.....	138
2) setRowheaderChecked.....	139

3) setRowheaderCheckedArr.....	139
4) getCheckedRows.....	140
14. Merge.....	141
14.1. header 영역.....	141
14.2. data 영역.....	142
15. Excel.....	143
1) excelDownload.....	143
2) excelUpload.....	145
3) excelDownloadURLHuge.....	146
4) exportCellInfoURLHuge.....	147
5) excelExportInfo.....	148
16. Group.....	150
1) setGroup.....	150
2) clearGroup.....	151
3) setGrandTotal.....	151
17. Contextmenu.....	153
1) setContextMenu.....	153
2) setDisableContextmenu.....	154
3) setAbleContextmenu.....	155
4) getContextmenuId.....	155
18. Event.....	156
1) bind.....	156
2) unbind.....	157
3) getUserEventInfo.....	158
4) trigger.....	158
5) clickCell.....	158
6) dblClickCell.....	159
7) focusCell.....	160
19. Extend.....	161
18.1 렌더링.....	161
18.2 서식 지정.....	162

18.3 빈 데이터 안내문구(emptymessage).....	163
5) setAbleContextmenu.....	163
20. 트리(Tree).....	164
1) setTree.....	164
2) insertTreeNode.....	166
3) removeTreeNode.....	168
4) updateTreeNode.....	169
5) getTreeState	170
6) clearTreeState	170
21. Filter.....	171
1) filter	171
2) filterEx.....	174
3) resetFilter	176
4) resetFilterEx.....	176
22. 엑셀 외 저장 방식.....	178
22.1. html.....	178
1) htmlDownload	178
22.2. txt.....	179
1) txtDownload.....	179
22. 헤더(Header)	180
23.1. 제목(title) 동적 변경.....	180
1) setHeaderCellValue.....	180
2) setHeaderCellValueEx	181
3) getHeaderCellValueEx.....	181
6 장. 스타일(CSS) 수정하기.....	182
1. CSS 경로와 파일	182
2. QCELL 의 구성	183
2.1 Header.....	183
1) colheader.....	183
2) rowheader.....	183
3) edgeheader	184

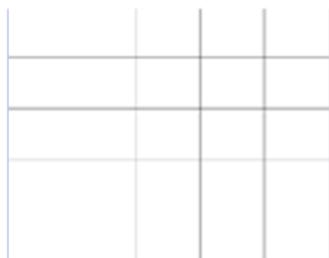
2.2	Viewport(Data)	184
1)	cell	185
2)	col, row	185
3)	frozencol, frozenrow, frozenbottomrow	186
2.3	기타	187
1)	move	187
2)	sort	187
3)	resize	187
4)	pagination	188

1 장. QCELL 이해하기

QCELL 은 웹 화면을 구성하는 여러 컴포넌트 중 가장 중요시하고 활용도가 가장 높은 컴포넌트로 그리드 기반의 데이터를 보여주거나 입력, 수정할때 사용하며 화면 로직 개발에서 차지하는 비중이 매우 높아 효율적이고 효과적인 개발이 필요한 영역입니다.

웹 그리드란

'그리드(Grid)'는 단어의 사전적 의미가 말해 주듯이 아래와 같은 격자무늬를 말합니다.



집안의 창문 모양에서도 볼 수 있고, 우리가 흔히 쓰는 엑셀(Excel) 프로그램과 워드(Word) / 파워포인트 (Powerpoint)의 '표' 와 같은 형태도 바로 이 그리드(Grid) 형태를 가지고 있습니다.

grid 미국·영국 [grɪd] 발음듣기 영국식 발음듣기 다른 뜻(3 건) 예문보기

1. 격자무늬
2. (금속·목재로 된) 격자판
3. (지도에서 위치를 나타내기 위한) 기준선망 (출처: 네이버)

시스템 개발과 웹 그리드

그럼 이러한 그리드가 수많은 SI 시스템 개발과 기업의 업무 현장에서 어떠한 연관성이 있는지 보겠습니다.

어떠한 기업 또는 기관이건 주제와 상관없이 다양한 '목록 (List)'을 작성하고 관리합니다. 고객 리스트 / 환자 처방전 리스트 / 장비 리스트 등등이죠. 또한 이러한 다양한 업무 및 고객 서비스 시스템은 관계형 데이터베이스(Database)를 기반으로 해서 데이터의 생성과 조회 / 수정 / 삭제가 빈번하게 발생합니다.

이러한 데이터를 화면에 보여주도록 하고 PC 의 엑셀처럼 각각의 데이터를 관리하기 위해 원하는 기능 (정렬 / 합계 등)을 수행할 수 있도록 해주는 UI 컴포넌트가 바로 그리드입니다. 그리고 업무 시스템 화면에서 엑셀의 화면 모습과 비슷한 그리드 화면 (표 / 리스트 등)을 보셨을 텐데 이러한 화면과 기능을 운영체제(OS)뿐만 아니라 PC /

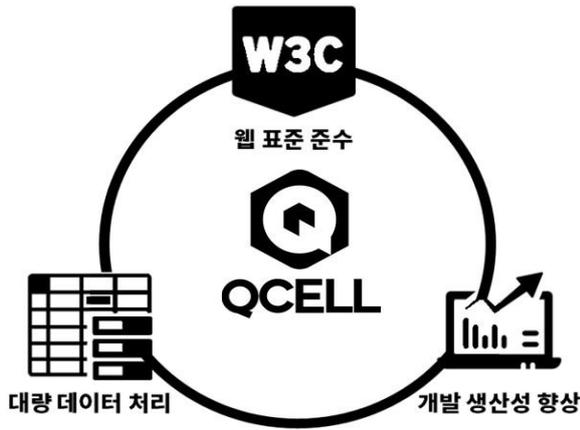
Mobile 에 비종속적인 환경으로 개발하기 위해 웹(브라우저)에서 동작하도록 만들어진 컴포넌트를 '웹 그리드 (Web grid)'라고 합니다.

그렇다 보니 기능은 예전의 C/S (클라이언트서버 Client Server) 환경이나 Active-X 기반의 UI 솔루션에서 지원하여 사용하고 있는 수준 그대로 웹 환경에서도 유지되기를 희망하고 있습니다. 어떠한 말이나 하면, 과거 Active-X 기반의 UI 솔루션 (기본적으로 그리드의 기능이 가장 핵심입니다)들은 웹 표준을 따르지 않고 그것을 만든 회사의 스펙에 따라 만들어지긴 했지만 그 기능들과 화면의 완성도는 굉장히 뛰어났습니다. 그런데 최근 몇 년 동안 HTML5 로 대변되는 웹 표준의 이슈가 거세게 불어닥치면서 기능과 화면의 완성도는 과거와 같이 유지되면서 웹 표준 환경을 따라주기를 바라는 요구 사항이 많아진 것입니다. 그러한 대표적인 기능들은 아래와 같이 데이터의 표현과 이용에 적합한 기능들입니다.

- 칼럼(Column) 헤더 / 로우 (Row) 헤더
- 데이터의 변경 / 갱신 / 복사 / 붙여넣기 / 추출
- 행의 상하단 고정 / 스타일 관리
- 컬럼 타입 설정 / 고정 / 넓이
- CRUD (생성 / 읽기 / 갱신 / 삭제) 처리
- 엑셀 연동
- 페이지 네이션 (Pagenation)
- 그룹핑 (Grouping)
- 검색 / 정렬
- 이벤트 (행위와 상태)

QCELL 소개

QCELL 은 대용량 데이터를 시각화 하기 위한 W3C 웹 표준을 준수하는 웹 그리드와 그리드 UI 개발 도구입니다.



Non-Active X 제품으로써 jQuery 기반의 순수 100% JavaScript 제품으로써 웹표준성을 준수하며, OSMU(One Source Multi Use)가 가능하여 어떤 OS, Device, Web Browser(IE8+, Chrome, Firefox, Opera, Edge 외 모바일 브라우저)에서도 동작 가능한 제품입니다.

특히 대용량 데이터 처리(좌우 및 상하 스크롤링의 사용자 상호작용)에 강점을 가지며, 개발 생산성 향상을 위한 Eclipse

Plugin 기반의 QEDITOR 를 통한 개발 Assist 를 지원합니다.

QCELL 은 PC 용 웹 화면뿐 아니라 모바일(탭, 스마트폰) 웹에서도 그리드를 지원하여 사용자의 미디어 타입(단말의 스크린 크기)에 반응형으로 생성하고 동작합니다.

기반 기술이 다른 그리드와 비교

그리드 모양이 격자 형태여야 하고 기능에 대한 대체적인 합의가 있을 뿐 어떤 기술(화면 정의, 데이터 정의, 로직 언어 등)을 사용해야 한다는 규정은 없습니다. 그렇기 때문에 오픈 소스 (Open source) 기반의 그리드나 상용 그리드 모두 각자가 선택한 기술을 사용하여 개발하였습니다.

웹 표준을 준수한다는 의미는 범용 웹 브라우저를 지원해야 한다는 것을 포함하고 있기 때문에 로직은 자바스크립트 (JavaScript)를 기반으로 개발되었으며, 가장 큰 차이는 그리드를 표현하는 화면 정의와 화면 처리입니다.

1. HTML Canvas tag 방식

화면을 Canvas 방식으로 그려주고, 그 안에서 발행하는 이벤트를 처리해 다시 그려주는 방식입니다. Canvas 특성상 그 안에 그려지는 화면은 HTML 코드로 표현되지 않아 웹브라우저가 제공하는 개발자 도구로 확인할 수 없습니다. 또한 특정한 로우(Row)를 선택했을 경우 해당 로우에 대해서만 HTML 화면 코드로 생성하여 보여줍니다. 이렇다 보니 Canvas 방식의 경우, 그리드 디자인(스타일)을 프로그래머가 프로그래밍 방식처럼 작업을 해야 하고 쉽게 바꿀 수 없으며, 나머지 화면 영역인 웹 표준 스타일 정의인 CSS 와 다른 체계로 관리하고 동작합니다.

웹브라우저에서 동작을 하지만, 일반적인 표현을 위한 태그로 동작하는 것이 아닌 그래픽 방식으로 처리되어, 엄격하게는 순수한 HTML 방식이라고 하기에는 무리가 있습니다.

```

<div style="position: relative; width: 100%; height: 100%; border-style: none; border-width: 0px; overflow: hidden; user-select: none; -webkit-tap-highlight-color: rgba(0, 0, 0, 0);">
  <canvas width="705" height="300" tabindex="0" role="application" style="position: absolute; left: 0px; top: 0px; background: rgb(255, 255, 255); border-style: none; border-width: 0px; cursor: default; -ms-filter: none; -ms-high-contrast-adjust: none; -ms-ime-mode: auto; -ms-text-orientation: mixed; -ms-user-select: none; -ms-zoom: 1; -webkit-backdrop-filter: none; -webkit-tap-highlight-color: transparent; -webkit-user-select: none; border: none; width: 0px; height: 0px; padding: 0px; margin: 0px; box-shadow: none; resize: none; left: 170px; top: 114px;">
  <div spellcheck="false" aria-hidden="true" style="outline: none; position: absolute; z-index: 2000; box-sizing: border-box; overflow: hidden; border: none; width: 0px; height: 0px; padding: 0px; margin: 0px; box-shadow: none; resize: none; left: 170px; top: 114px;">
    <input aria-hidden="true" autocomplete="off" tabindex="0" id="realgrid_line" maxlength="1000000" style="position: absolute; margin: 0px; padding: 0px; font-family: Tahoma; font-style: normal; font-variant: normal; font-weight: normal; font-size: 10pt; line-height: normal; overflow-wrap: normal; overflow: hidden; resize: none; border: none; outline: none; text-align: left; color: rgb(68, 84, 106); left: 0px; top: 0px; width: 100%; height: 100%; cursor: text; -webkit-user-modify: read-write-plaintext-only; white-space: pre-wrap; transform: translateZ(0px); text-transform: none;">
  </div>
</div>

```

	✓	Order ID	Customer ID	Employee ID	Order Date	Company Name
1	<input type="checkbox"/>	10248	VINET	5	1996/07/04	Vins et alcools Chevalier
2	<input type="checkbox"/>	10248	VINET	5	1996/07/04	Vins et alcools Chevalier
>	<input type="checkbox"/>	10248	VINET	5	1996/07/04	Vins et alcools Chevalier
4	<input type="checkbox"/>	10249	TOMSP	6	1996/07/05	Toms Spezialitäten
5	<input type="checkbox"/>	10249	TOMSP	6	1996/07/05	Toms Spezialitäten

위처럼 활성화되는 영역이 발생할 때 이 부분은 그래픽으로 처리할 수 없어서 Canvas 태그 하단에 동적으로 html 태그로 표현하는 방식으로 처리합니다. 위의 그림에서 입력하는 input element 하나가 Canvas 위에 그려지는 방식입니다.

즉, 화면을 렌더링하고 화면에 접근하고 처리하는 방식이 일반적인 웹 표준을 준수하고 있지 않은 구조입니다. 이벤트에 따라 Canvas의 그래픽을 처리할 것인지, html 태그를 처리할 것인지 결정되는 구조로, 화면에 대한 동작으로 프로그래머가 파악하기 어려운 구조입니다.

2. HTML Table tag 방식

격자와 동일한 모양이라 빠르게 그리드 솔루션을 개발할 수 있으며, 기본적인 기능 등을 Table의 처리로 활용할 수 있습니다. 그래서 많은 솔루션들이 이 방식으로 화면을 정의하고 있습니다.

그러나, 표현해야 할 컬럼과 로우만큼의 화면을 그려야 하고, 보이지 않는 영역까지 처리하는 것이 부담이었습니다. 그래서 화면의 이동 처리가 빠르지 못한 점이 있습니다.

```

<table cellpadding="0" id="SBHE_DT_datagrid" class="sbgrid_table sbgrid_data_table_st sbgrid_DT sbgrid_DT_st sbgrid_common" style="width: 1100px;
<tbody class="sbgrid_common">
  <tr data-rowindex="1" class="sbgrid_common" style="height: 25px;">
    <td class="sbgrid_cell sbgrid_cell_border_st sbgrid_cell_st sbgrid_data_cell_st sbgrid_datagrid_col_col_0_style sbgrid_checkbox_data_col_st sbgrid_ellipsis sbgrid_common" data-rowindex="1" data-colindex="0" cell-type="3">
    <td class="sbgrid_cell sbgrid_cell_border_st sbgrid_cell_st sbgrid_data_cell_st sbgrid_datagrid_col_col_1_style sbgrid_input_data_col_st sbgrid_ellipsis sbgrid_common" data-rowindex="1" data-colindex="1" cell-type="3" col-hidden="none" style="display: none;"> 50
      <span class="sbgrid_span sbgrid_common" style="white-space: nowrap;">청량도우린이발기술학원</span>
    <td class="sbgrid_cell sbgrid_cell_border_st sbgrid_cell_st sbgrid_data_cell_st sbgrid_datagrid_col_col_2_style sbgrid_output_data_col_st sbgrid_ellipsis sbgrid_common" data-rowindex="1" data-colindex="2" cell-type="3">
    <td class="sbgrid_cell sbgrid_cell_border_st sbgrid_cell_st sbgrid_data_cell_st sbgrid_datagrid_col_col_3_style sbgrid_input_data_col_st sbgrid_ellipsis sbgrid_common" data-rowindex="1" data-colindex="3" cell-type="3" col-hidden="none" style="display: none;">
    <td class="sbgrid_cell sbgrid_cell_border_st sbgrid_cell_st sbgrid_data_cell_st sbgrid_datagrid_col_col_4_style sbgrid_textarea_data_col_st sbgrid_ellipsis sbgrid_common" data-rowindex="1" data-colindex="4" cell-type="3">
    <td class="sbgrid_cell sbgrid_cell_border_st sbgrid_cell_st sbgrid_data_cell_st sbgrid_datagrid_col_col_5_style sbgrid_combo_data_col_st sbgrid_ellipsis sbgrid_common" data-rowindex="1" data-colindex="5" cell-type="3">
    <td class="sbgrid_cell sbgrid_cell_border_st sbgrid_cell_st sbgrid_data_cell_st sbgrid_datagrid_col_col_6_style sbgrid_combo_data_col_st sbgrid_ellipsis sbgrid_common" data-rowindex="1" data-colindex="6" cell-type="3">
    <td class="sbgrid_cell sbgrid_cell_border_st sbgrid_cell_st sbgrid_data_cell_st sbgrid_datagrid_col_col_7_style sbgrid_datepicker_data_col_st sbgrid_ellipsis sbgrid_common" data-rowindex="1" data-colindex="7" cell-type="3">
    <td class="sbgrid_cell sbgrid_cell_border_st sbgrid_cell_st sbgrid_data_cell_st sbgrid_datagrid_col_col_8_style sbgrid_inputdate_data_col_st sbgrid_ellipsis sbgrid_common" data-rowindex="1" data-colindex="8" cell-type="3">
    <td class="sbgrid_cell sbgrid_cell_border_st sbgrid_cell_st sbgrid_data_cell_st sbgrid_datagrid_col_col_9_style sbgrid_outputbutton_data_col_st sbgrid_ellipsis sbgrid_common" data-rowindex="1" data-colindex="9" cell-type="3">
    <td class="sbgrid_cell sbgrid_cell_border_st sbgrid_dummy_cell_st sbgrid_common" data-colindex="-1">
  </tr>

```

상임자(성명)	학원주소	교습과정	교습과목(반)	span.sbgrid span.sbgrid_common	63.58
박영배	서울특별시동대문구왕산로33길32층(제기동)	이미용	남성	2017-03-31	
대상희계실무학원(주)	서울특별시동대문구왕산로225.5층2호, 5층6호(정랑리동)	종합(대)	여성	2017-03-31	
대상희계실무학원(주)	서울특별시동대문구왕산로225.5층2호, 5층6호(정랑리동)	종합(대)	A	2017-03-31	
김병규이종운	서울특별시동대문구고산자로36길33층일부(제기동)	전자	자동차기기	2017-03-31	
김종환	서울특별시동대문구이문로70(이문동)	음악	음악조금	2017-03-31	
(주)에너지엔트코리아	서울특별시동대문구왕산로312층3호(서성동)	자동차	검사	2017-03-31	

```

  <td class="sbgrid_cell sbgrid_cell_border_st sbgrid_cell_st sbgrid_data_cell_st sbgrid_datagrid_col_col_4_style sbgrid_textarea_data_col_st sbgrid_ellipsis sbgrid_common" data-rowindex="1" data-colindex="4" cell-type="3">
  <td class="sbgrid_cell sbgrid_cell_border_st sbgrid_cell_st sbgrid_data_cell_st sbgrid_datagrid_col_col_5_style sbgrid_combo_data_col_st sbgrid_ellipsis sbgrid_common" data-rowindex="1" data-colindex="5" cell-type="3">
  <td class="sbgrid_cell sbgrid_cell_border_st sbgrid_cell_st sbgrid_data_cell_st sbgrid_datagrid_col_col_6_style sbgrid_combo_data_col_st sbgrid_ellipsis sbgrid_common" data-rowindex="1" data-colindex="6" cell-type="3">
  <td class="sbgrid_cell sbgrid_cell_border_st sbgrid_cell_st sbgrid_data_cell_st sbgrid_datagrid_col_col_7_style sbgrid_datepicker_data_col_st sbgrid_ellipsis sbgrid_common" data-rowindex="1" data-colindex="7" cell-type="3">
    <span class="sbgrid_span sbgrid_common" style="white-space: nowrap;">2017-03-31</span>
  </td>
  <td class="sbgrid_cell sbgrid_cell_border_st sbgrid_cell_st sbgrid_data_cell_st sbgrid_datagrid_col_col_8_style sbgrid_inputdate_data_col_st sbgrid_ellipsis sbgrid_common" data-rowindex="1" data-colindex="8" cell-type="3">
    <table class="sbgrid_inputdate sbgrid_common" cellpadding="0">
      <thead>
        <tr>
          <th colspan="2" class="sbgrid_common"></th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <td style="width: 80%;" class="sbgrid_common"></td>
          <td style="width: 20%;" class="sbgrid_common"></td>
        </tr>
      </tbody>
    </table>
  </td>
  <td class="sbgrid_cell sbgrid_cell_border_st sbgrid_cell_st sbgrid_data_cell_st sbgrid_datagrid_col_col_9_style sbgrid_outputbutton_data_col_st sbgrid_ellipsis sbgrid_common" data-rowindex="1" data-colindex="9" cell-type="3">
    <td class="sbgrid_cell sbgrid_cell_border_st sbgrid_dummy_cell_st sbgrid_common" data-colindex="-1">
  </tr>
  <tr data-rowindex="2" class="sbgrid_datagrid_alternate_style sbgrid_common" style="height: 25px;">

```

이렇게 눈에 보이지 않는 부분에 대하여 계속해서 처리해야 하고, 컬럼이 많아질수록 이

부분에 대한 부담이 매우 큰 화면 처리 기술을 사용합니다.

3. HTML Div tag 방식

Div tag 를 하나의 셀(Cell)로 만들고, 이를 모아 컬럼과 로우로 만들어 격자 모양의 그리드를 표현하는 것으로, Table 에 비해 표현의 자유도가 높고, 좌우 스크롤 시 테이블과 달리 보여지는 영역만 바뀌는 방식으로 위의 두 가지 방식보다 빠르게 화면의 이동 처리가 가능합니다. 웹 표준을 준수하면서 데이터 표현과 화면 스타일링, 빠른 화면 처리가 가능하도록 최근에 고안되었으며, 저희의 QCELL(큐셀) 또한 이 방식을 적용하고 있습니다.

```

<div id="sheetarea" style="width: 100%; height: 550px;">
  <div class="rt-qc-root" id="root-qcell" style="width: 100%; height: 100%;">
    <div class="rt-qc-panel" id="panel-qcell" style="width: 1221px; height: 0px; display: none;"></div>
    <div class="rt-qc-paginationpanel" id="paginationpanel-qcell" style="width: 1221px; height: 0px; top: 892px; display: none;"></div>
    <div class="rt-qc-container" id="container-qcell" tabindex="0" style="width: 1221px; height: 548px;">
      <div class="rt-qc-viewport" id="viewport-qcell" style="height: 451px; width: 1172px; left: 50px; top: 80px;">
        <div class="rt-qc-viewport-inner" tabindex="0" style="width: 1172px; height: 451px;">
          <div class="rt-qc-cell rt-qc-cell-viewport rt-qc-cell-frozencols align-center rt-qc-cell-normalrows rt-qc-static-type-cell" tabindex="0" data-row="2" data-col="1" id="cell_2_1" style="width: 100px; height: 37px; line-height: 35px; left: 0px; top: 0px;">2018년 12월</div>
          <div class="rt-qc-cell rt-qc-cell-viewport rt-qc-cell-normalcols aligncenter rt-qc-cell-normalrows rt-qc-html-type-cell" tabindex="0" data-row="2" data-col="2" id="cell_2_2" style="width: 80px; height: 37px; line-height: 35px; left: 100px; top: 0px; padding-right: 10px; text-align: right;"></div>
          <div class="rt-qc-cell rt-qc-cell-viewport rt-qc-cell-normalcols aligncenter rt-qc-cell-normalrows rt-qc-html-type-cell" tabindex="0" data-row="2" data-col="3" id="cell_2_3" style="width: 80px; height: 37px; line-height: 35px; left: 180px; top: 0px; padding-right: 10px; text-align: right;"></div>
          <div class="rt-qc-cell rt-qc-cell-viewport rt-qc-cell-normalcols aligncenter rt-qc-cell-normalrows rt-qc-html-type-cell" tabindex="0" data-row="2" data-col="4" id="cell_2_4" style="width: 80px; height: 37px; line-height: 35px; left: 260px; top: 0px; padding-right: 10px; text-align: right;"></div>
          <div class="rt-qc-cell rt-qc-cell-viewport rt-qc-cell-normalcols aligncenter rt-qc-cell-normalrows rt-qc-html-type-cell" tabindex="0" data-row="2" data-col="5" id="cell_2_5" style="width: 80px; height: 37px; line-height: 35px; left: 340px; top: 0px; padding-right: 10px; text-align: right;"></div>
          <div class="rt-qc-cell rt-qc-cell-viewport rt-qc-cell-normalcols aligncenter rt-qc-cell-normalrows rt-qc-html-type-cell" tabindex="0" data-row="2" data-col="6" id="cell_2_6" style="width: 80px; height: 37px; line-height: 35px; left: 420px; top: 0px; padding-right: 10px; text-align: right;"></div>
          <div class="rt-qc-cell rt-qc-cell-viewport rt-qc-cell-normalcols aligncenter rt-qc-cell-normalrows rt-qc-html-type-cell" tabindex="0" data-row="2" data-col="7" id="cell_2_7" style="width: 80px; height: 37px; line-height: 35px; left: 500px; top: 0px; padding-right: 10px; text-align: right;"></div>
          <div class="rt-qc-cell rt-qc-cell-viewport rt-qc-cell-normalcols aligncenter rt-qc-cell-normalrows rt-qc-html-type-cell" tabindex="0" data-row="2" data-col="8" id="cell_2_8" style="width: 80px; height: 37px; line-height: 35px; left: 580px; top: 0px; padding-right: 10px; text-align: right;"></div>
          <div class="rt-qc-cell rt-qc-cell-viewport rt-qc-cell-normalcols aligncenter rt-qc-cell-normalrows rt-qc-html-type-cell" tabindex="0" data-row="2" data-col="9" id="cell_2_9" style="width: 80px; height: 37px; line-height: 35px; left: 660px; top: 0px; padding-right: 10px; text-align: right;"></div>
          <div class="rt-qc-cell rt-qc-cell-viewport rt-qc-cell-normalcols aligncenter rt-qc-cell-normalrows rt-qc-html-type-cell" tabindex="0" data-row="2" data-col="10" id="cell_2_10" style="width: 80px; height: 37px; line-height: 35px; left: 740px; top: 0px; padding-right: 10px; text-align: right;"></div>
          <div class="rt-qc-cell rt-qc-cell-viewport rt-qc-cell-normalcols aligncenter rt-qc-cell-normalrows rt-qc-html-type-cell" tabindex="0" data-row="2" data-col="11" id="cell_2_11" style="width: 80px; height: 37px; line-height: 35px; left: 820px; top: 0px; padding-right: 10px; text-align: right;"></div>
        </div>
      </div>
    </div>
  </div>

```

2 장. 구성요소

화면에 보여질 QCELL(웹 그리드)의 구성 요소의 이름을 먼저 이해하고 숙지해야 분석, 설계, 개발시 프로젝트에 참여한 구성원들이 동일하게 용어를 사용하여 진행할 수 있습니다.

1. cell

cell(셀)은 그리드에 데이터를 표현하는 최소 단위이며 한 칸(격자)을 의미합니다. 일반적으로는 데이터 영역의 셀을 지칭하며 아래의 모든 격자는 하나의 셀이며 그 셀들이 모여 큐셀이라는 그리드를 구성합니다.

cell(셀)

	<input type="checkbox"/>	firstname	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
9	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
10	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
11	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
13	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

cell(셀)

2. row 와 column(col)

큐셀(그리드)는 셀들이 모여서 만들어진 모습으로 일반적으로 데이터베이스 테이블과 유사하게 같은 데이터 속성을 가진 셀들을 묶어서 지칭하거나 셀들이 하나의 객체나 데이터를 의미하도록 묶어서 지칭해야 하는 경우가 많습니다.

- **row(로우, 행)**는 cell 들을 가로 단위로 묶은 단위의 명칭입니다.
- **column(컬럼, 열)** 이란 cell 들을 세로로 묶은 단위의 명칭이며 코딩하기 편리하도록 row 와 맞춰 줄여서 col 이라고 사용합니다.

column(컬럼, 열)

	<input type="checkbox"/>	firstname	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
9	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
10	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
11	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
13	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

row(로우, 행)

3. frozenrow, frozenbottomrow 와 frozencol

큐셀은 경우에 따라 많은 셀들(로우와 컬럼)을 화면에 보여줘야 하는 경우가 있습니다. 이때 항상 보여주고 싶은 로우나 컬럼을 정의해야 하는데 이때 사용하는 용어가 틀고정입니다. 큐셀이 지원하는 틀고정은 크게 상단 로우, 하단 로우, 좌측 컬럼입니다.

- **frozenrow(상단 로우 틀고정)**은 헤더 바로 밑의 로우를 기준하여 몇 개의 행을 고정시켜 상하 스크롤이 발생할 때에도 계속해서 보여줄 것인지를 의미합니다.
- **frozenbottomrow(하단 로우 틀고정)**은 맨 아래의 로우를 기준으로 상하 스크롤 발생과 무관하게 처음 그리드가 생성될때부터 보여줄 로우 수를 의미합니다.
- **frozencol(좌측 컬럼 틀고정)**은 큐셀의 컬럼이 많아서 화면에 모든 컬럼이 나오지 않게 되면 좌우 스크롤이 나타납니다. 이때 좌우 스크롤을 하더라도 좌측의 필요한 컬럼은 항상 보일 수 있도록 설정할 수 있는데, 이때의 설정값이 frozencol 입니다.

frozencol
(컬럼 틀고정)

	<input type="checkbox"/>	firstname	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
9	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
10	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
11	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
13	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

frozenrow
(상단 로우 틀고정)

frozenbottomrow
(하단 로우 틀고정)

그림은 상단 2 개의 로우, 하단 1 개 로우, 좌측 2 개 컬럼을 틀고정한 모습이며, 틀고정된 로우와 컬럼임을 사용자가 쉽게 식별할 수 있도록 스타일(셀 배경)이 다르게 표시됩니다. 또한 화면에 모든 컬럼과 로우가 보여질때는 틀고정의 의미가 없는 상황이므로 틀고정 스타일이 반영되지 않습니다.

4. scrollbar

웹 화면의 포함되어 생성되는 큐셀은 그 크기가 정해져서 만들어집니다. 이때 큐셀에 생성되어 보여지는 셀들(로우, 컬럼)이 많으며 그리드 영역에 모두 표현할 수 없게 넘쳐버리게 됩니다. 이때 넘쳐버린 셀들을 볼 수 있도록 사용자에게 제공하는 것이 스크롤바입니다.

- **Vertical scrollbar(상하 스크롤바)**는 큐셀로 표현할 수 있는 로우수가 넘쳐날 때 발생하는데 항상 발생하는 것은 아니고 큐셀에 바인딩되어 보여줄 데이터 건수에 따라 달라집니다.
- **Horizontal scrollbar(좌우 스크롤바)**는 큐셀의 화면 영역으로 컬럼을 표현할 수 없어 넘쳐날 때 발생합니다. 웹 브라우저의 가로 넓이에 따라 달라집니다.

Vertical scrollbar
(상하 스크롤바)

	<input type="checkbox"/>	firstname	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
9	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
10	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
11	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
13	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

Horizontal scrollbar
(좌우 스크롤바)

3 장. 준비와 설치하기

큐셀을 사용하여 웹 화면에 그리드를 개발하기 위한 절차를 설명하려고 합니다. 1 장에서 설명드렸듯이 JavaScript 기반의 라이브러리 제품이므로 이 라이브러리를 그리드가 필요한 화면에서 사용할 수 있도록 만들어 주어야 합니다. 이 과정이 설치이며 먼저 웹 서버의 리소스 폴더에 라이브러리를 복사하고 이 라이브러리를 로딩할 수 있도록 웹 프로그래밍 환경을 구성합니다.

이러한 특징이 있어서 서버의 어떤 운영체제 환경, 서버 개발 언어, 웹 브라우저의 종류에 상관없이 동작할 수 있습니다.

1) QCELL 라이브러리

QCELL 은 동적으로 그리드를 웹 화면내에 그려주기 위해 동작하는 JavaScript 와 그려진 그리드에 보여지는 스타일과 이미지로 구성된 라이브러리입니다.

라이브러리 폴더의 구조는 스타일과 폰트 관련 내용이 있는 css 폴더, 참조하는 이미지 파일들이 있는 img 폴더 그리고 주요 JavaScript 라이브러리들이 있는 js 폴더로 구성되어 있습니다.

qcell.js 와 **qcell.min.js** 은 QCELL 의 핵심 JavaScript 파일입니다.

- 1) **qcell.js**: QCELL 을 사용하기 위한 환경설정 파일
- 2) **qcell.min.js**: QCELL 의 각종 속성 및 구동을 하기 위한 파일

2) QCELL 설치

2.1 시스템 요구 사항

시스템 요구사항은 서버와 클라이언트로 나눌 수 있으며, 서버는 JavaScript 를 포함하여 웹 개발할 수 있는 환경이면 됩니다. 실질적으로 모든 웹 개발 서버 환경에서 사용할 수 있습니다.

클라이언트 환경은 그리드를 활용한 데이터의 크기나 로직의 복잡도에 따라서 더 큰 메모리와 빠른 CPU 가 필요할 수 있으며 JavaScript 기반의 동작이므로 스크립트 실행이 빠른 브라우저에서 사용하는 것을 권장합니다.

제품	항목	최소 사양	권장 사양
QCELL 2.0	CPU	400MHz Intel, 호환 CPU	800MHz Intel, 호환 CPU
	HDD	제약 없음	
	메모리	최소 1.5GB RAM	4GB RAM
	운영체제	Any OS(MS Windows, Linux 계열, OSX)	
	웹 브라우저	IE 9.0 이상	IE 11.0 이상
	다양한 브라우저	크롬, 사파리, 파이어폭스, 오페라 지원 가능	

2.2 QCELL 제품 구성

소프트웨어 제품에 대한 납품이므로 큐셀 라이브러리는 외에 사용자 매뉴얼, API 문서, 개발자 툴킷, 라이선스 증서가 함께 제공됩니다.

1) 라이브러리

QCELL 라이브러리는 마지막 릴리즈 된 날짜와 버전명으로 이뤄진 압축 파일로 제공됩니다.

예: [2020-07-17]QCELL_v2.1.98.zip

2) 사용자 매뉴얼

QCELL 사용자 매뉴얼은 본 파일로 큐셀의 이해부터, 구성, 설치 방법, 큐셀 생성과 기능 등 사용에 대한 전반적인 설명을 포함하고 있습니다. 매뉴얼에서 이례를 돕기 위해 포함된 예제도 사용해 볼 수 있도록 함께 제공합니다.

3) API 문서

QCELL API 문서는 QCELL 이 제공하는 API 에 대한 설명과 예제 소스를 포함하고 있으며 웹 페이지 형태로 제공됩니다. 기술사이트를 통해서도 서비스하고 있습니다.

4) 개발자 툴킷

큐셀을 처음 사용하는 개발자들이 큐셀을 이해 후 좀 더 빨리 효율적으로 원하는 기능을 구현할 수 있도록 바로 활용할 수 있는 샘플들을 개발자 눈높이에서 만들어 제공하는 패키지입니다.

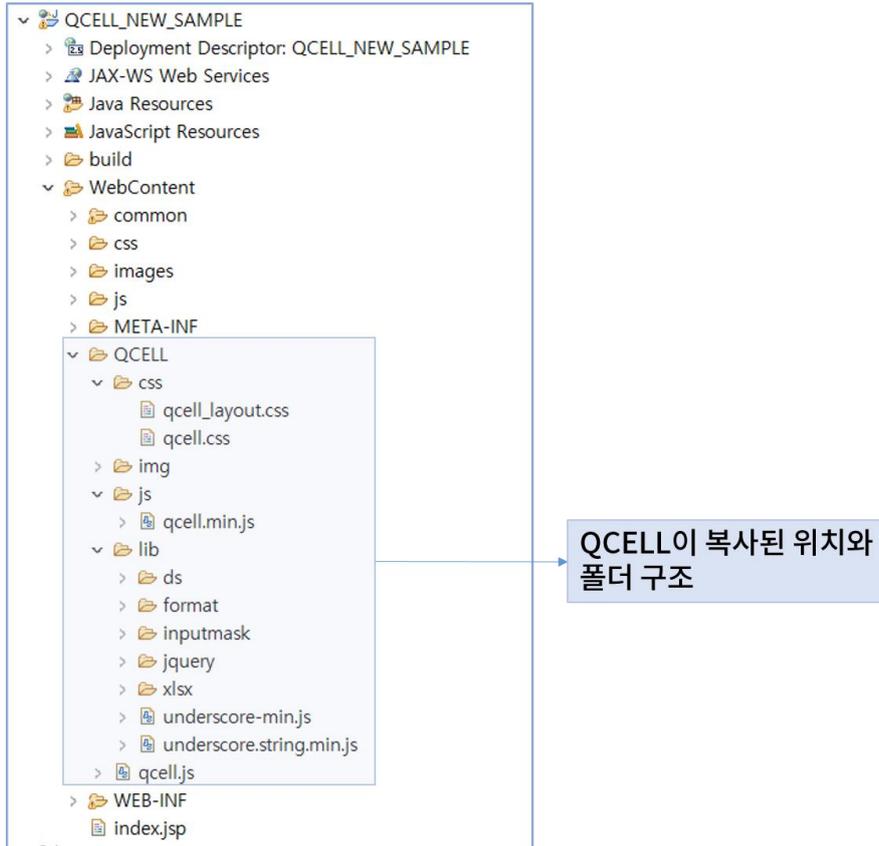
5) 라이선스 증서

QCELL 제품 구입 시 아래의 그림과 같은 형태로 정품 인증서가 제공됩니다. 이는 추후 기술지원을 받을 때 필요합니다.



2.3 QCELL 패키지 복사

앞서 설명드렸듯이 QCELL의 설치는 웹 서버의 적절한 폴더에 압축하여 제공한 라이브러리를 압축 해제 후에 복사하고 웹 프로젝트에서 해당 경로를 포함시켜 개발합니다.



2.4 QCELL 라이브러리 설치 확인하기

정상적으로 설치되어 화면 개발 시 큐셀을 생성할 수 있는 환경인지 확인하는 절차로 프로젝트 하위에 html 파일을 생성한 후 아래와 같은 순서로 QCELL 을 사용하기 위해 라이브러리를 로딩합니다.

- 1) QCELL CSS 호출

```
5 <link href="./QCELL/css/qcell.css" type="text/css" rel="stylesheet" />
```

- 2) QCELL 경로지정

QCELL 의 경로를 JavaScript 변수 RightTechPath 에 설정합니다. 이 경로를 통해 QCELL 이 필요로 하는 다양한 라이브러리(Javascript) 파일들을 로딩합니다. 이 경로가 올바르지 못하면 QCELL 은 동작하지 않습니다.

QCELL 라이브러리가 복사된 부모 폴더를 가르켜야 합니다. 즉, 이 Path(경로)에 QCELL 폴더가 위치하고 있으면 됩니다.

```
6 <script type="text/javascript">
7     var RightTechPath = './';
8 </script>
```

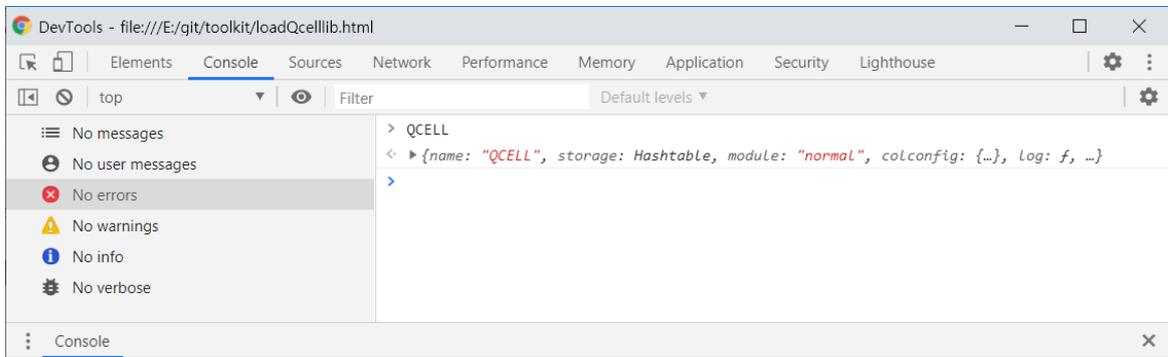
3) QCELL 설정 파일 호출

QCELL의 경로를 올바르게 지정하였다면, QCELL의 설정 파일을 호출하는 것으로 QCELL 라이브러리 로딩을 실시합니다. 이렇게 로딩하고 나면 이 화면에서는 큐셀을 만들 수 있는 준비가 된 것입니다.

```
9 <script type="text/javascript" src="./QCELL/qcell.js"></script>
```

4) QCELL 설치 확인

QCELL 라이브러리가 정상적으로 호출되었는지 개발자도구의 콘솔 창에서 "QCELL" 네임스페이스를 입력하여 아래와 같이 확인하면 됩니다.



check eXample 툯킷 내 loadQcelllib.html 참조

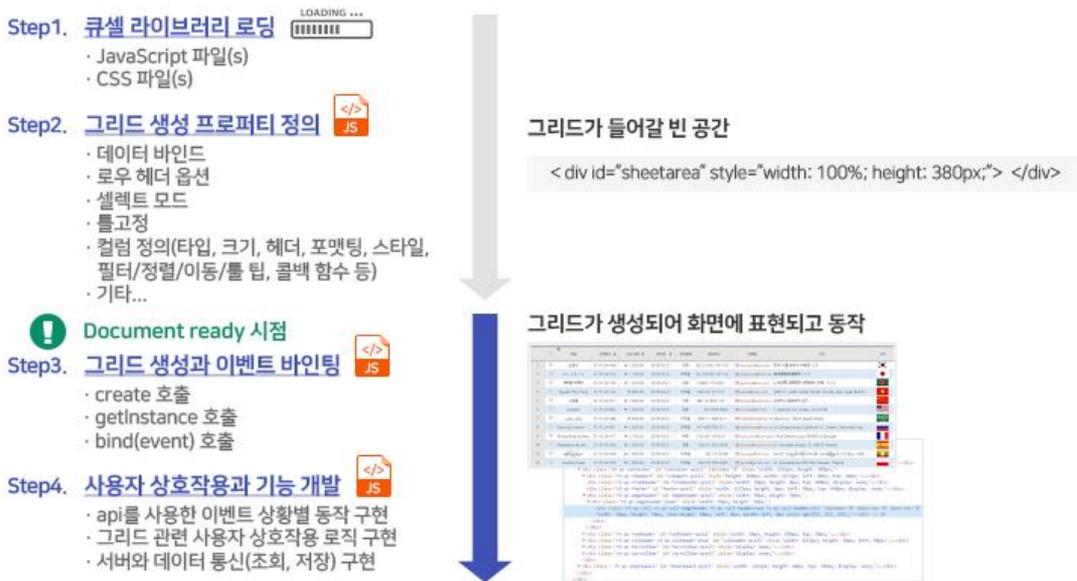
아래는 jsp 파일에서 로딩하는 형태의 예제입니다.

```
12 <link href='<c:url value="/QCELL/css/qcell.css"/>' type="text/css" rel="stylesheet" />
13 <script type="text/javascript">
14     var RightTechPath = '/';
15 </script>
16 <script type="text/javascript" src='<c:url value="/QCELL/qcell.js"/>'></script>
```

4 장. 생성하기

1. QCELL 생성하기

QCELL을 생성하는 과정은 아래 그림과 같으며 먼저 라이브러리를 로딩하고 생성하고 싶은 그리드를 속성 값을 사용하여 설정합니다. 이 속성들 중에 그리드가 만들어져 들어갈 부모 컨테이너 아이디를 설정해야 합니다. 설정을 모두 완료 후 그리드를 생성하고 생성된 그리드의 인스턴스 값을 얻어서 원하는 이벤트 바인딩과 기능을 QCELL이 제공하는 API를 사용하여 프로그래밍 할 수 있습니다.



1.1. 부모 컨테이너 선언

- 1) 먼저 화면 내에 그리드를 위치할 곳과 크기(넓이와 높이)를 결정한 후 꼭 div 태그로 정의하고 이때 id를 정의해야 하며 이 값을 그리드 생성할 프로퍼티 paraentid에 값으로 설정해야 합니다.

이때 높이는 px 만 가능하고 넓이는 px, % 모두 가능합니다.

```
64 <div id="sheetarea" style="height: 100px; width: 100%;"></div>
```

- 2) 이 div 태그 안에 그리드가 생성되므로 부모 컨테이너라고 합니다. 부모 컨테이너 크기(넓이와 높이) 안에서 그리드가 그려짐을 숙지해야 합니다. 이때 컬럼을 넣을 공간이 부족하면 좌우 스크롤바가 생기고 그리드에 바인딩되는 데이터 건수를 높이 안에서 표현할 수 없으면 상하 스크롤바가 자동으로 생겨서 모든 컬럼과 데이터로 이동하고 볼 수 있습니다.

1.2. QCELL 속성 설정

QCELL을 생성할 때 사용할 수 있는 프로퍼티는 그리드 용도와 기능에 따라 값을 설정해야 하는데 아래와 같이 최소한의 프로퍼티 값 설정으로도 큐셀을 생성할 수 있습니다.

- 1) 간단한 설정으로 QCELL을 생성할 수 있습니다.

아래처럼 qcell id, qcell 이 그려질 부모 컨테이너 id 그리고 그리드에 들어갈 컬럼을 1 개 이상 정의하면 됩니다. 컬럼 정의는 컬럼 헤더에 보일 컬럼명과 컬럼의 넓이(폭), 컬럼에 보여줄 데이터 키 값만 정의하면 됩니다.

```

17     var QCELLProp = {
18       id : 'qcell',
19       parentid : 'sheetarea',
20       columns : [
21         {key: 'name', title: ['컬럼1'], width: "20%"}
22         , {key: 'message', title: ['컬럼2'], width: "30%"}
23         , {key: 'etc', title: ['컬럼3'], width: "50%"}
24       ]
25     }

```

- 2) QCELL 의 종속 라이브러리로 jQuery(1.12.4)를 사용하고 있어 별도의 jQuery 라이브러리를 호출할 필요가 없습니다. (타 버전의 jQuery 를 사용해야 하는 경우 QCELL 라이브러리 호출 후 사용할 버전의 jQuery 를 호출하면 됩니다. 이 경우 충돌을 회피하기 위하여 QCELL 은 별도의 네임으로 jQuery 를 정의하여 **✓check eXample** 사용합니다.)

[FAQ 내 "jQuery 버전 충돌이 발생하지 않지요?" 참조](#)

- 3) "QCELL.create()" API 를 호출하여 QCELL 을 생성합니다. 인자로 정의한 프로퍼티를 넘겨주면 parented 영역에 속성 값으로 그려집니다.
- 한 화면에 부모 컨테이너를 여러 개 정의하고 여러 개의 그리드를 생성할 수 있습니다. 이때 parentid 는 모두 달라야 합니다.
 - 생성된 큐셀을 이용하여 프로그래밍 하기 위해서는 생성되는 QCELL 객체가 각각 다른 id 를 가지고 있어야 합니다.

```

27     QCELL.create(QCELLProp);

```

- 4) QCELL 에 표시할 컬럼의 속성들을 "columns" 속성으로 설정하는데 QCELL 은 매우 많은 것을 제공하고 있습니다. 그 외에도 그리드 전체에 대한 설정을 다양하게 할 수 있는데 좀 더 자세한 설명은 다음 절 또는 API 문서를 참고합니다.

✓check eXample [API 문서내 QCELL.create\(\) 의 properties 표 참조](#)

- 5) "QCELL.getInstance()" 함수를 호출하여 QCELL 객체를 획득합니다.

객체를 얻어내야 하는 이유는 큐셀을 그리고 끝나는 것이 아니고 그리드에 이벤트를 등록하거나 화면의 동작에 따라 그리드를 조작하여 사용자가 원하는 기능을 수행해야 하는데 이때 그리드를 가르치는 인스턴스가 있어야 이 인스턴스를 사용하여 API 를 사용할 수 있습니다.

```

28     qcell = QCELL.getInstance("qcell");

```

지금까지 입력한 코드는 툴킷의 소스를 참조하시기 바랍니다.

이제 가장 간단하게 QCELL 을 만들기 위한 html 화면을 코딩 하였습니다. 이제 저장 후 웹 브라우저 실행하여 이 파일을 실행(접근)하면 아래와 같이 QCELL(그리드)이 생성됩니다.

컬럼1	컬럼2	컬럼3

큐셀에 보여줄 샘플 데이터를 아래와 같이 선언하고 프로퍼티에 data 속성에 선언한 값을 바인딩 해주고 다시 웹 브라우저로 실행해 보면 아래와 같이 데이터가 나옵니다.

```
13      var sampleData = [{"name":"Hello","message":"큐셀!","etc":"가장 간단한 큐셀 그리드입니다."}];

17      var QCELLProp = {
18          id : 'qcell',
19          parentid : 'sheetarea',
20          data : {'input': sampleData},
```

컬럼1	컬럼2	컬럼3
Hello	큐셀!	가장 간단한 큐셀 그리드입니다.

 **check eXample** 툴킷 내 [helloqcell.html](#) 참조

2. QCELL Property

자세한 내용과 최신 버전이 제공하는 속성은 라이브러리와 함께 제공하는 API 문서를 참조해 주시기 바랍니다. 주요한 프로퍼티(속성)와 값 설정 방법과 의미를 정리한 것입니다.

2.1 QCELL Property

큐셀의 전체 영역에 대한 부분이고 컬럼별 프로퍼티는 다음 절에 있습니다.

명칭	타입	설명
parenteid	string	QCELL 을 표시할 div 컨테이너의 ID
id	string	생성할 QCELL 객체의 ID
data	Object	표시할 데이터(JSON Array 타입)
rowheader	string	표시할 행 번호 타입(default: none) ("none" "sequence" "reverse")

selectmode	string	선택 단위 (default: cells) ("rows" "row" "cols" "col" "cells" "cell")
highlightrepeat	number	행 스타일을 반복하여 적용하는 "highlightrepeat" 속성의 행 반복 주기 (default:0)
frozencols	number	틀 고정 열 개수 (default:0)
frozenrows	number	틀 고정 행 개수 (default:0)
frozenbottomrows	number	하단 틀 고정 행 개수 (default:0)
rowheight	Object	행 높이
rowheight.header	number	헤더 영역의 높이(단위: pixel, default: 40)
rowheight.data	number	데이터 표시 영역의 높이 (단위: pixel, default: 40)
copy	boolean	ctrl+c 를 이용하여 선택된 셀들에 대한 클립보드 복사 기능 사용 여부
paste	boolean	클립보드 데이터를 ctrl+v 를 이용하여 그리드에 붙여넣기 기능 사용 여부
oneclickedit	boolean	input 계열 컬럼 사용시 마우스 한 번 클릭으로 editmode 상태로 되는 여부 설정 (default: false)
useGroupUI	boolean	GROUP UI 사용 여부
excelexportwithformat	boolean	엑셀 export 시에 format 포함한 적용 값 여부
bottompanel	Boolean	QCELL 하단에 excel 처럼 선택한 셀들의 갯수, 합계, 평균을 표시하시는 패널 표시 여부
bottompanelheight	number	QCELL 하단 패널 영역의 높이
pagination	object	pagination 설정
pagination.pageunit	number	화면에 출력 설정이 가능한 Data 의 Row 수
pagination.unitlist	array	화면에 출력 설정이 가능한 Data 의 Row 수의 배열
pagination.url	string	server 와 통신할 url 주소
pagination.totalsort	boolean	pagination 사용시 url 주소가 설정되어 있을 때 server 와 통신을 처음 한 번만 할 것인가 선택

pagination.param	object	url 주소가 설정되어 있을 때 전달하고자 하는 파라미터 정보
pagination.loadonce	boolean	url 주소가 설정되어 있을 때 server 와 통신을 처음 한 번만 할 것인가 선택 (default: false)
pagination.totalsort	boolean	pagination 사용시 url 주소가 설정되어 있을 때
pagination.mode	string	pagination 사용 방식 설정 (default: "simple") ("simple" "extend")
pagination.pagecount	number	pagination mode 'extend' 일 때 하단에 표시할 페이지 갯수 설정 (default: 5)
pagination.extendmove	boolean	pagination mode 'extend' 일 때 하단에 이동할 페이지 입력 화면 표시 (default: false)
pagination.useProgress	boolean	pagination url 일 경우 서버 통신 중 Progress 표시 여부
onloadurl	string	QCELL 생성시 server 와 통신할 URL 주소
emptymessage	string	데이터 없을 때 표시할 text
hiddencolumns	array	Hidden 컬럼에 대한 정보를 저장할 수 있는 Property
merge	object	병합 방법(JSON Array 타입)
merge.header	string	헤더 영역의 병합 방식 (default: "none") ("none" "row" "col" "rowandcol" "colandrow")
merge.data	string	데이터 영역의 병합 방식 (default: "none") ("none" "row" "col" "group")
contextmenu	object	생성할 contextmenu 정보
contextmenu.header	object	header 영역에서 생성할 contextmenu 의 정보
contextmenu.header.id	string	contextmenu 의 ID
contextmenu.header.contents	array	contextmenu 구성 정보
contextmenu.header.contents.icon	string	아이콘으로 사용할 이미지의 경로 fontawesome 의 i Tag 지원

contextmenu. header.contents.label	string	출력할 메뉴명
contextmenu. header.contents.callback	function string	메뉴 클릭시 동작할 콜백 함수
contextmenu. header.contents.items	array	현재 메뉴에서 생성할 하위 메뉴 정보
contextmenu. header.contents.divine	boolean	메뉴 분리 표시 여부
contextmenu. header.contents.disabled	boolean	메뉴 비활성화 여부

contextmenu.header 를 사용한 contextmenu 예시

[Syntax & Example]

```

contextmenu :
{
  header : {
    id : "header-contextmenu",
    contents:
    [
      {icon: '<i class="fas fa-plus"></i>', label: '메뉴 header-a', callback: function(){qcell.addRow();}},
      {divine: true},
      {icon: '<i class="fas fa-minus"></i>', label: '메뉴 header-b', callback: function(){qcell.removeRow();}}
    ]
  }
}

```

이름	이메일	사명	입사일	주소
고길동 010-1234-5678	gogildong@gamil.com	길동전자	1978-02-03	서울 특별시 송파구
둘리 010-2345-6789	twori@gamil.com	길동전자	1992-07-21	서울 특별시 송파구

contextmenu.data	object	data 영역에서 생성할 contextmenu 의 정보
------------------	--------	--------------------------------

contextmenu.data.base	boolean	date 영역에서 기본적으로 생성할 contextmenu 의 정보
contextmenu.data.base.id	string	contextmenu 의 ID
contextmenu.data.base.contents	Array	contextmenu 구성 정보
contextmenu.data.base.contents.icon	string	아이콘으로 사용할 이미지의 경로(fontawesome 의 I Tag 지원)
contextmenu.data.base.contents.label	string	출력할 메뉴명
contextmenu.data.base.callback	function string	메뉴 클릭시 동작할 콜백 함수
contextmenu.data.base.items	Array	현재 메뉴에서 생성할 하위 메뉴 정보
contextmenu.data.base.divine	Boolean	메뉴 분리 표시 여부
contextmenu.data.base.disabled	Boolean	메뉴 비활성화 여부

contextmenu.data.base 를 사용한 contextmenu 예시

[Syntax & Example]

```
contextmenu:
{
  data: {
    base: {
      id: 'base-contextmenu',
      contents:
      [
        { icon: '<i class="fas fa-plus"></i>', label: '메뉴 base-a', callback: function () { qcell.addRow(); } },
        { icon: '<i class="fas fa-minus"></i>', label: '메뉴 base-b', callback: function ()
{ qcell.removeRow(); } },
        { divine: true }
      ]
    }
  }
}
```

```

    ]
  },
}
}

```

이름	전화번호	이메일	사명	입사일	주소
고길동	010-1234-5678	gogildong@gamil.com <div style="border: 1px solid black; padding: 2px; width: fit-content; margin-left: 20px;"> 메뉴base-a 메뉴base-b </div>	길동전자	1978-02-03	서울 특별시 송파구
둘리	010-2345-6789	twori@gamil.com	길동전자	1992-07-21	서울 특별시 송파구

contextmenu.data.except	boolean	date 영역에서 예외적으로 생성할 contextmenu 의 정보
contextmenu.data.except.id	string	contextmenu 의 ID
contextmenu.data.except.col	number	data 영역에서 예외적으로 생성할 column 의 index
contextmenu.data.except.contents	Array	contextmenu 구성 정보
contextmenu.data.except.contents.icon	string	아이콘으로 사용할 이미지의 경로(fontawesome 의 I Tag 지원)
contextmenu.data.except.contents.label	string	출력할 메뉴명
contextmenu.data.except.callback	function string	메뉴 클릭시 동작할 콜백 함수
contextmenu.data.except.items	Array	현재 메뉴에서 생성할 하위 메뉴 정보
contextmenu.data.except.divine	Boolean	메뉴 분리 표시 여부
contextmenu.data.except.disabled	Boolean	메뉴 비활성화 여부

contextmenu.data.except 를 사용한 contextmenu 예시

[Syntax & Example]

```

contextmenu:
{
  data: {
    except: [
      {
        col: 3,
        id: 'except1-contextmenu',
        contents: [
          {
            label: 'except1-d', items:
            [
              { label: 'except1-d-1-1', callback: "objClick" },
              { label: 'except1-d-1-2', callback: "objClick" }
            ]
          }
        ]
      },
    ]
  }
}

```

이름	전화번호	이메일	사명	입사일	주소
고길동	010-1234-5678	gogildong@gamil.comm	길동전자		
둘리	010-2345-6789	twori@gamil.comm	길동전자	1992-07-21	서울 특별시 송파구

except1-d >

- except1-d-1-1
- except1-d-1-2



2.2 QCELL Columns Property

컬럼에 대한 프로퍼티이며 어떤 타입으로 값을 정의하고 해당 정의에 대한 의미는 무엇인지 나타내는 표입니다.

명칭	타입	설명
key	string	컬럼에 표시할 key 값 (표시할 데이터의 타입이 "array"인 경우 "number"타입으로 설정)
type	string	컬럼 타입 (default: static) ("static" "input" "textarea" "datepicker" "selectmenu" "checkbox" "image")
styleclassname	object	컬럼 스타일 클래스 설정
styleclassname.header	string	헤더 영역 스타일 클래스 이름
styleclassname.data	string	데이터 영역스타일 클래스 이름
title	string or array	컬럼의 헤더 타이틀 (멀티 타이틀 설정 가능)
width	string	컬럼 width 설정 속성('%' 사용 가능) (default:100px)
datatype	string	컬럼의 데이터 타입 설정 속성 (default: string) ('string' or 'number')
move	boolean	컬럼 이동 기능 사용 여부(default: false)
sort	boolean	컬럼 정렬 기능 사용 여부(default: false)
resize	boolean	컬럼 너비 조절 기능 사용 여부(default: false)
search	boolean	컬럼 검색 기능 사용 여부(default: false)
tooltip	boolean	컬럼 툴팁 기능 사용 여부
disabled	boolean	"static", "html" 컬럼 타입 외의 모든 컬럼 타입 비활성화 여부 설정
editor	boolean	컬럼의 스타일을 editor 를 통해 설정 여부
style	object	컬럼 스타일 설정

style.header	object	헤더 영역 스타일 정보
style.data	object	데이터 영역 스타일 정보
options	object	컬럼타입 의존 속성 정보
options.required	Boolean	필수 입력 컬럼 여부
options.mergecol	Boolean	merge type "col" 이용시 컬럼의 merge 실행 여부
options.format	object	"static", "input" 컬럼타입 사용시 컬럼의 데이터에 대한 포맷
options.format.type	string	"static", "input" 컬럼타입 사용시 포맷 타입 ("string" "number" "date" "custom")
options.format.origin	string	"static", "input" 컬럼타입 사용시 데이터의 원본 날짜 포맷 ("date" 포맷 타입에서만 사용, ex> "yyyy-mm-dd" "dd-mm-yy")
options.format.rule	string	"static", "input" 컬럼타입 사용시 데이터를 출력할 포맷 ("custom" 포맷 타입 생략 가능, ex> " <code>₩ #,###.##</code> ") * 상세한 format 설정은 https://momentjs.com/docs/#/parsing/string-format/ 을 참조
options.format.locale	string	"static", "input" 컬럼타입 사용시 "type"이 "number" 나 "date"일 때 사용하려는 국가코드 입력, ex> "en" "kr"
options.format.callback	function	"static", "input" 컬럼타입 사용시 "custom" 포맷 타입에서만 사용이 가능하며, 커스터마이징 한 format 함수
options.format.allowempty data	boolean	빈 값일때도 동작할 수 있는 속성. 기본 동작의 변경이며 명시적으로 속성값을 true 로 설정해야지만 동작 가능
options.html	object	"static" 컬럼타입 사용시 컬럼의 각 영역(헤더, 데이터)별 renderer 정보
options.html.header	function	컬럼의 헤더 영역 renderer 콜백 함수
options.html.data	function	컬럼의 데이터 영역 renderer 콜백 함수

options.mask	object	"input" 컬럼타입 사용시 입력 형태 표시
options.mask.alias	string	"input" 컬럼타입 사용시 입력 형태 (ex> "(999)-9999-9999")
options.numrange	string	"input" 컬럼타입 사용시 입력 가능한 숫자 범위 지정
options.wordwrap	boolean	데이터를 줄바꿈 형태로 표시(상단정렬 고정)
options.maxlength	number	"input", "textarea" 컬럼타입 사용시 최대 입력 가능 수
options.limit	object	"input", "textarea" 컬럼타입 사용시 입력제한 방식 설정
options.limit.type	string	입력제한 타입 설정 ("all" "string" "number" "other") * number 는 '0 부터 9 와 ',' 사용가능, string 은 '0 부터 9'를 제외한 나머지 사용가능, other 는 입력가능한 문자 설정 가능
options.limit.regexp	string	입력제한 타입이 "other" 일 때 사용, 입력 가능한 문자 설정
exclude	Array	사용자가 입력 후 제거해야 할 문자 들 설정 ex ["",\W] *
options.textareawidth	string	"textarea" 컬럼타입 사용시 textarea 의 넓이
options.locale	string	"datepicker" 컬럼타입 사용시 출력할 달력의 언어 ("ko" "en") (default: "ko")
options.dateformat	string	"datepicker" 컬럼타입 사용시 원본 데이터의 날짜 포맷 (default: " yy.mm.dd ") (ex> "yy.mm.dd" "ddmmyy")
options.changeyear	boolean	"datepicker" 컬럼타입 사용시 달력의 년 선택 가능 여부 (default: false)
options.changemonth	boolean	"datepicker" 컬럼타입 사용시 달력의 월 선택 가능 여부 (default: false)
options.yearrange	string	"datepicker" 컬럼타입 사용시 달력 년도 선택 가능 기간 설정 (ex> "c-1000:c+1000" "+100:-100" "1900 2200")
options.tooltipcallback	function	원하는 동작, 메시지 등을 지정할 수 있는 툴팁 콜백함수(개발툴킷 059 번 참조)

options.timetype	string	"datepicker" 컬럼타입 사용시 timepicker 의 사용 여부
options.timeformat	string	"datepicker" 컬럼타입 사용시 timepicker 의 시간 포맷 (ex> "h:m:s") H 는 24 시간, h 는 12 시간, m 은 분, s 는 초이며, HH 와 같이 두 글자로 설정한 경우에만 앞에 0 을 표시
options.dateimage	string	"datepicker" 컬럼타입 사용시 컬럼에 출력할 calendar 의 이미지 주소 (이미지의 크기는 기본 16x16 으로 출력)
options.dateimageside	string	"datepicker" 컬럼타입 사용시 calendar 의 좌우위치 설정 ("left" "right")
options.dateimageonly	Boolean	"datepicker" 컬럼타입 사용시 click 에 대한 동작을 calendar 에만 할 것인지 설정 여부
options.input	array	"selectmenu" 컬럼타입 사용시 사용할 콤보 아이템 JSON Object 의 레퍼런스
options.itemcount	number	"selectmenu" 컬럼타입 사용시 표현할 콤보 아이템의 표시 개수
options.selectedEnd	Boolean	"selectmenu" 컬럼타입 사용시 아이템을 선택하면 종료의 설정 여부 (bind("selected")로 이벤트를 설정할 수 있으며, parameter 는 (event, row, col, item)이다.)
options.selectclickopen	boolean	"selectmenu" 컬럼타입 사용시 editmode 활성화시 selectmenu 펼쳐침 설정 여부
options.wholeselect	boolean	checkbox 컬럼 타입, 전체 선택 checkbox 생성 설정 (true or false) - likeradio 가 true 일 때 미 동작 (default: false)
options.checkedvalue	string or boolean	checkbox 컬럼 타입 선택 시 데이터 값 설정 속성 (ex> 'true') (default: true)
options.uncheckedvalue	string or boolean	"checkbox" 컬럼타입 사용시 선택 해제했을 때의 데이터의 값 (default: false)
options.likeradio	boolean	"checkbox" 컬럼타입 사용시 동작을 "radio"와 같이 하나만 선택 가능 (default: false)
options.imagealt	string	"image" 컬럼타입 사용시 image element 의 대체 문구

options.imageheight	string	"image" 컬럼타입 사용시 image element 의 높이
options.imagewidth	string	"image" 컬럼타입 사용시 image element 의 넓이
options.imagestyle	string	"image" 컬럼타입 사용시 image element 의 스타일
options.imageclick	string	"image" 컬럼타입 사용시 image element 를 클릭할 때 동작할 event 콜백 함수
options.imageuri	string	"image" 컬럼타입 사용시 출력할 image 의 경로
options.imageerroruri	string	"image" 컬럼타입 사용시 에러가 발생할 경우 출력할 에러 image 의 경로
min	number	"number", "slider" 컬럼타입 사용시 표현할 값의 최소값 설정
max	number	"number", "slider" 컬럼타입 사용시 표현할 값의 최대값 설정
step	number	"number", "slider" 컬럼타입 사용시 변경할 값의 범위
buttonwidth	string	"button" 컬럼타입 사용시 적용할 버튼의 넓이
buttonheight	string	"button" 컬럼타입 사용시 적용할 버튼의 높이
buttonposition	string	"button" 컬럼타입 사용시 적용할 버튼의 위치
buttonlabel	string	"button" 컬럼타입 사용시 적용할 버튼의 라벨
buttonclick	function	"button" 컬럼타입 사용시 버튼 클릭 시 동작할 함수

5 장. 기능 이해하기

1. QCELL 생성/제거하기

웹 화면의 부모 컨테이너에 QCELL 을 생성하고 제거하는 것에 대하여 알아보겠습니다. 제거는 해당 영역에 설정을 달리한 큐셀을 생성하고 싶은 때 먼저 제거를 하고 생성해야 합니다. 그렇지 않고 특정한 큐셀을 만들고 계속 사용하는 경우 제거를 하지 않아도 됩니다. 화면이 전환될 경우 로딩된 라이브러리와 DOM 등이 웹 브라우저에 의해 제거됩니다.

1) create

create 는 QCELL 의 객체를 생성하는 API 입니다.

Syntax & eXample

```
(static) create(properties) → {void}
```

properties: QCELL 생성시 설정하는 속성 <필수> : [object]

```

15 QCELL.create({
16   "parentid"      : "QCellArea",
17   "id"            : "QCell1",
18   "data"          : {"input": cell_data},
19   "rowheader"    : "sequence",
20   "selectmode"   : "cell",
21   "highlightrepeat" : 2,
22   "frozencols"   : 2,
23   "frozenrows"   : 3,
24   "frozenbottomrows" : 2,
25   "rowheight"    : {header: 40, data: 40},
26   "oneclickedit" : false,
27   "onloadurl"    : "URL",
28   "columns"      : [
29     {title:["순번"], key:"seq"},
30     {title:["선택"], key:"checkbox", options: {checkedvalue: "ok", uncheckedvalue: "ng"}},
31     {title:["이름"], key:"name", type:"input", maxlength: 4},
32     {title:["나이"], key:"age", type:"input", options: {format: {type: "string", rule: "(
33     {title:["연락처"], key:"phone", type:"input", options: {mask: {alias: "999-9999-0
34     {title:["주소"], key:"address", type:"textarea"},
35     {title:["일사일"], key:"regdate", type:"datepicker ", options:{locale:"kr", datefo
36   ]},
37   "pagination"   : {pageunit: 10, unitlist: [10, 20, 30]},
38   "contextmenu"  : {
39     base: {
40       id: 'base-contextmenu',
41       contents:
42       [
43         {icon: './add.png', label: 'a', action: function(){QCell1.addRow();}},
44         {icon: './del.png', label: 'b', action: function(){QCell1.removeRow();}},
45         {label: 'c', items: [{label: 'c-1'}, {label: 'c-2'}]},
46         {label: 'd', items: [{label: 'd-1', items: [{label: 'd-1-1'}, {label: 'd-1-2'}]}, {label: 'd-2', ite
47       ]
48     }
49   }
50 });

```

2) hasInstance

현재 화면에 입력한 ID 에 해당하는 QCELL 객체가 생성되어 있는지를 확인하는 API 입니다.

 Syntax & eXample

```
(static) hasInstance(id) → {Boolean}
```

id: 존재 여부를 반환할 QCELL 객체의 id 설정 <필수> : [string]

```
30 QCELL.hasInstance("sheet1");
```

3) getInstance

생성한 QCELL 그리드 객체를 얻어야 API 를 통해 원하는 동작을 실행시킬 수 있습니다. 먼저 생성된 큐셀 객체를 얻을 때 사용하는 API 입니다. 이때 입력하는 id 는 create 인자인 프로퍼티 내의 id 값이어야 합니다.

 Syntax & eXample

```
(static) getInstance(id) → {Object}
```

id: 반환할 QCELL 객체의 id 설정 <필수> : [string]

```
32 var sheet1 = QCELL.getInstance("sheet1");
```

4) getInstanceIDs

화면에 하나 이상의 QCELL 이 create 된 상태일 때 모든 객체를 얻어내어 원하는 동작을 실행하고 싶을 때 모든 객체를 배열로 얻어내는 API 입니다.

 Syntax & eXample

```
(static) getInstanceIDs() → {Array}
```

인자는 없으며, 결과는 객체 값들의 배열

```
30 var arrSheetIDs = QCELL.getInstanceIDs();
```

5) deleteInstance

생성된 QCELL 객체를 지우는 API 입니다. 기본적으로 QCELL 객체는 화면이 전환되거나 갱신되면 자동으로 지워지는데 임의로 지우고 다른 QCELL 객체로 생성하는 경우엔 반드시 지워야 해당 부모 컨테이너에 새로운 객체를 생성할 수 있습니다.

 Syntax & eXample

```
(static) deleteInstance(id) → {Boolean}
```

id: 제거할 QCELL 객체의 id 설정 <필수> : [string]

```
32 QCELL.deleteInstance("sheet1");
```

6) deleteAllInstances

화면에 생성되어 있는 모든 QCELL 객체를 지우는 API 입니다.

 Syntax & eXample

```
(static) deleteAllInstances() → {Boolean}
```

```
34 QCELL.deleteAllInstances();
```

2. 초기화

생성된 QCELL 객체를 필요에 따라 다시 만들거나, 화면 만을 다시 그리는 동작이 필요하여 API 를 제공하고 있습니다.

1) rebuild

동적 스타일 적용, 컬럼 이동, 틀고정 등의 변화된 모습에서 원래 초기 프로퍼티 값의 QCELL 로 다시 만들고 싶을 때 사용하는 API 입니다.

 Syntax & eXample

```
rebuild() → {boolean}
```

```
40 QCell1.rebuild();
```

 Do programming

```

45     var property = {
46         id       : "qcell",
47         parentid : "sheetarea",
48         columns  :
49         [
50             {title: ["firstname"], key: "firstname", sort: true},
51             {title: ["lastname"], key: "lastname"},
52             {title: ["age"], key: "age"},
53             {title: ["phone"], key: "phone"},
54             {title: ["birthday"], key: "birthday"},
55             {title: ["company"], key: "company"},
56             {title: ["city"], key: "city"},
57             {title: ["street"], key: "street"},
58             {title: ["address"], key: "address"},
59             {title: ["registered"], key: "registered"}
60         ],
61         rowheaders : ["sequence", "checkbox"],
62         data       : {"input": check_data},
63         frozenrows  : 0,
64         frozenbottomrows : 0,
65         frozencols  : 0
66     }
67
68     function createQCELL(){
69         QCELL.create(property);
70         qcell = QCELL.getInstance("qcell");
71     }
72
73     function rebuild(){
74         qcell.rebuild()
75     }
76
77     function changeStyle(){
78         qcell.setColStyle(3, {"background-color" : "#aadd11"} );
79         qcell.setRowStyle(3, {"background-color" : "#aadd11"});
80     }

```

 See result

- rebuild 실행 전

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Av
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac
3	<input checked="" type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac
4	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac
5	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac
6	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac

- rebuild 실행 후

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Av
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac
4	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac
5	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac
6	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac

 **check eXample** 전체 소스는 manual 내 "rebuild API 동작" 참조

2) refresh

단순히 현재 보여지고 있는 화면 영역을 갱신하는 API 입니다.

Syntax & eXample

```
refresh() → {boolean}
```

```
42 QCell1.refresh();
```

3) resetData

QCELL 내의 데이터를 수정, 삭제 등이 실시하여 데이터가 변경된 상태에서 초기 바인딩된 데이터로 다시 적용하여 그리드를 만들 때 사용하는 API 입니다.

Syntax & eXample

```
resetData() → {boolean}
```

```
45 QCell1.resetData();
```

4) initGrid

QCELL 그리드를 초기화하는 API 입니다.

 Syntax & eXample

`initGrid()` → {boolean}

QCELL 객체의 ID.initGrid()
Ex> `qcell.initGrid()`;

3. Property & Index

QCELL 을 핸들링 하기 위해서는 객체 ID 를 얻어야 하고 해당 객체 ID 룡 로우, 컬럼 수와 현재 선택된 셀 등의 인덱스를 할 수 있어야 데이터를 얻거나 변경할 수 있고 스타일도 설정할 수 있기에 가장 기본이 되는 API 입니다.

1) getID

QCELL 의 ID 를 반환하는 API 입니다. ID 별로 동작해야 하는 경우 로직 내에서 QCELL 의 객체에 대한 ID 를 얻어서 비교하는 경우에 사용됩니다.

 Syntax & eXample

`getID()` → {string}

QCELL 의 original 프로퍼티가 true 일 때 사용 가능

```
42      var strID = QCell1.getID();
```

2) getIdx

QCELL 은 입력 받은 조건에 해당하는 행(또는 열)인덱스를 반환하는 API 입니다.

 Syntax & eXample

`getIdx(axis, typeopt, point_of_timeopt)` → {number}

axis: 인덱스 종류 <필수> : [string]

row	로우 인덱스
col	컬럼 인덱스

type: 인덱스 타입 <선택> : [string]

cursor	커서
drag	드래그

	(dragging 중에 인덱스 값을 확인해야 하는 경우에만 의미있는 값이며 "mousemove" 이벤트를 바인딩하여 확인해야 합니다.)
click	클릭
focus(default)	포커스

point_of_time: 인덱스 시점 <선택> : [string]

previous	이전 위치 (이전 값이 존재하는 경우에만 의미있는 값이며 그리드 생성 후 셀을 처음 선택 시에는 이전 값이 없으므로 getIdx("row", "focus", "previous")의 반환 값이 -1 이 정상입니다. 다른 셀을 다시 선택 후 확인해보면 정상적으로 인덱스 값을 확인 할 수 있습니다.
current(default)	현재 위치

```

47 QCell1.getIdx("row"); //포커스 셀의 행 인덱스 반환
48 QCell1.getIdx("col"); //포커스 셀의 열 인덱스 반환
49 QCell1.getIdx("row", "cursor"); //커서가 위치한 셀의 행 인덱스 반환
50 QCell1.getIdx("col", "click"); //마우스 왼쪽 클릭이 발생한 열 인덱스 반환

```

 Do programming

```

34 function getIdx(){
35     nRow = qcell.getIdx("row");
36     nCol = qcell.getIdx("col");
37     nHeaderRow = qcell.getRows("header");
38     if(nRow < nHeaderRow){
39         alert("셀을 선택하여 주십시오.");
40     } else {
41         alert("row:" + nRow + ", col:" + nCol);
42     }
43 }

```

 See result

- 3,5 cell 클릭 후 getIdx() 확인

getIdx로 선택한 셀의 인덱스(로우, 컬럼) 확인하기

	<input type="checkbox"/>	firstname	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Av
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-7...	2015/09/28	COGENTRY	Greenock	Tech Plac
4	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac
5	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac
6	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac

이 페이지 내용:

row:3, col:5

확인

 **check eXample** 전체 소스는 manual 내 "선택한 셀의 인덱스 얻기(getIdx API 동작 확인)" 참조

3) getRows

QCELL 은 입력 받은 영역에 해당하는 행의 개수를 반환하는 API 입니다.

 **Syntax & eXample**

```
getRows(typeopt) → {number}
```

type: 영역 구분 <선택> : [string]

all(default)	전체 행
header	헤더영역
data	헤더영역을 제외한 데이터를 표시하는 영역
frozen	틀 고정된 영역
frozenbottom	틀 고정된 하단 영역

```
34 QCell1.getRows(); //전체 행의 개수 반환
35 QCell1.getRows("data"); //헤더영역을 제외한 데이터를 표시하는 전체 행의 개수 반환
36 QCell1.getRows("frozen"); //틀고정 된 행(FrozenRows)의 개수 반환
```

 **Do programming**

```

48     function getRows(){
49         alert( "All rows: "+qcell.getRows());
50     }
51
52     function getDataRows(){
53         alert( "Data rows: "+qcell.getRows('data'));
54     }

```

 See result

전체 로우 수 얻기		데이터 영역의 로우 수 얻기							
	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Av
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac
4	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac
5	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac
6	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Plac

이 페이지 내용:
All rows: 14

이 페이지 내용:
Data rows: 13

 **check eXample** 전체 소스는 manual 내 "전체 로우 수, 데이터 영역의 로우 수를 얻기(getRows API 동작 확인)" 참조

4) getCols

QCELL 은 입력 받은 영역에 해당하는 컬럼의 개수를 반환하는 API 입니다.

 **Syntax & eXample**

```
getCols(typeopt) → {number}
```

type: 영역 구분 <선택> : [string]

all(default)	전체 행
header	헤더영역

data	헤더영역을 제외한 데이터를 표시하는 영역
frozen	틀 고정된 영역

```
38 QCell1.getCols(); //전체 열의 개수 반환
39 QCell1.getCols("data"); //헤더영역을 제외한 데이터를 표시하는 전체 열의 개수 반환
40 QCell1.getCols("frozen"); //틀고정 된 열(FrozenRows)의 개수 반환
```

5) getRowHeight

입력 받은 영역에 대한 로우의 높이를 반환하는 API 입니다.

Syntax & eXample

```
getRowHeight(type) → {number}
```

type: 영역 구분 <선택> : [string]

header	헤더영역
data	헤더영역을 제외한 데이터를 표시하는 영역

```
47 QCell1.getRowHeight("header");
```

See result

```
> qcell.getRowHeight("header");
< 40
> qcell.getRowHeight("data");
< 40
> |
```

6) getColWidth(s)

getColWidth 는 할당된 넓이를 알고 싶은 컬럼의 인덱스를 인자로 전달하고 실행하는 것이고 getColWidths 는 전체 컬럼들의 넓이 값을 배열로 반환하는 API 입니다.

Syntax & eXample

```
getColWidths() → {Array}
```

```
42 QCell1.getColWidths();
```

See result

```
> qcell.getColWidths()
< ▶ (12) [50, 50, 100, 100, 100, 100, 100, 100, 100, 100, 100, 100]
> qcell.getColWidth(1)
< 50
> |
```

4. 데이터

4.1. 데이터 얻기(반환), 갱신하기

1) getData

QCELL 객체에 바인딩 되어 그리드에 보여지고 있는 전체 데이터를 반환(얻어내는)하는 API 입니다.

 Syntax & eXample

```
getData() → {Array}
```

```
44 QCell1.getData();
```

 See result

m-0002.html 을 개발자 도구에서 api 실행한 결과

```
> qcell.getData()
< ▼ (13) [{"...", {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}, {...}]
  ▶ 0: {check1: false, check2: "ok", check3: "0", isActive: true, balance: "4,896,634.48", ...}
  ▶ 1: {check1: true, check2: "ng", check3: "1", isActive: false, balance: "3,567,933.28", ...}
  ▶ 2: {check1: true, check2: "ng", check3: "1", isActive: false, balance: "3,567,933.28", ...}
  ▶ 3: {check1: true, check2: "ng", check3: "1", isActive: false, balance: "3,567,933.28", ...}
  ▶ 4: {check1: true, check2: "ng", check3: "1", isActive: false, balance: "3,567,933.28", ...}
  ▶ 5: {check1: true, check2: "ng", check3: "1", isActive: false, balance: "3,567,933.28", ...}
  ▶ 6: {check1: true, check2: "ng", check3: "1", isActive: false, balance: "3,567,933.28", ...}
  ▶ 7: {check1: true, check2: "ng", check3: "1", isActive: false, balance: "3,567,933.28", ...}
  ▶ 8: {check1: true, check2: "ng", check3: "1", isActive: false, balance: "3,567,933.28", ...}
  ▶ 9: {check1: true, check2: "ng", check3: "1", isActive: false, balance: "3,567,933.28", ...}
  ▶ 10: {check1: true, check2: "ng", check3: "1", isActive: false, balance: "3,567,933.28", ...}
  ▶ 11: {check1: true, check2: "ng", check3: "1", isActive: false, balance: "3,567,933.28", ...}
  ▶ 12: {check1: true, check2: "ng", check3: "1", isActive: false, balance: "3,567,933.28", ...}
  length: 13
  ▶ __proto__: Array(0)
```

2) getAllData

모든 데이터를 반환하는 API 입니다.

 Syntax & eXample

```
getAllData() → {Array}
```

QCELL 객체의 ID.getAllData()

Ex> `qcell.getAllData();`

이름	전화번호	이메일	사명	입사일	주소
홍길동	010-11...	qwer@gmail.com	길동상가	20190813	서울특별시 송파구...
임걱정	010-22...	qwer@gmail.com	길동상가	20190813	서울특별시 송파구...
심청이	010-44...	qwer@gmail.com	길동상가	20190813	서울특별시 송파구...
문익점	010-66...	qwer@gmail.com	길동상가	20190813	서울특별시 송파구...
황건적	010-88...	qwer@gmail.com	길동상가	20190813	서울특별시 송파구...

```
> qcell.getAllData()
< ▼ (5) [{...}, {...}, {...}, {...}, {...}] ⓘ
  ▶ 0: {name: '홍길동', phone: '010-1111-2222', email: 'qwer@gmail.com', company: '길동상가', r
  ▶ 1: {name: '임걱정', phone: '010-2222-3333', email: 'qwer@gmail.com', company: '길동상가', r
  ▶ 2: {name: '심청이', phone: '010-4444-5555', email: 'qwer@gmail.com', company: '길동상가', r
  ▶ 3: {name: '문익점', phone: '010-6666-7777', email: 'qwer@gmail.com', company: '길동상가', r
  ▶ 4: {name: '황건적', phone: '010-8888-9999', email: 'qwer@gmail.com', company: '길동상가', r
    length: 5
```

3) getAllDataWithFormat

QCELL 은 모든 데이터를 format 이 적용된 형태로 반환하는 API 입니다.

Syntax & eXample

`getAllDataWithFormat() → {Array}`

QCELL 객체의 ID.getAllDataWithFormat()

Ex> `qcell.getAllDataWithFormat ();`

4) setData

QCELL 객체에 반영된 데이터를 입력한 데이터로 갱신(다른 데이터 셋을 반영)하는 API 입니다.

Syntax & eXample

`setData(data) → {boolean}`

data: 갱신할 데이터 <필수> : [Array]

```
30 //QCELL의 data 타입이 "object"인 경우
31 QCell1.setData([{"name" : "kevin", "age" : "35"}, {"name" : "tom", "age" : "21"}]);
32
33 //QCELL의 data 타입이 "array"인 경우
34 QCell1.setData(["kevin", "35"], ["tom", "21"]);
```

 Do programming

```
31 function setData() {
32     qcell.setData([{"age":30,"firstname":"James","lastname":"Kwak","company":"Apple","city":
33         {"age":40,"firstname":"Tomas","lastname":"Park","company":"Google","city":"Pusan
34
35 }
```

 See result

setData 확인하기

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	st
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech P
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech P
4	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech P
5	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech P
6	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech P

setData 확인하기 버튼 클릭으로 setData 실행 결과를 확인해 보면 데이터 건수가 작아지면서 상하 스크롤바는 사라지고 좌우 스크롤바만 반영되는 것을 볼 수 있습니다.

setData 확인하기

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	James	Kwak	30			Apple	Seoul	
2	<input type="checkbox"/>	Tomas	Park	40			Google	Pusan	

 check eXample 전체 소스는 manual 내 "QCELL의 데이터 변경하기(setData API 동작 확인)" 참조

5) getRowData

QCELL 객체에 반영된 데이터 내에서 지정한 로우의 데이터를 객체로 반환(얻어내는)하는 API 입니다.

Syntax & eXample

```
getRowData(row) → {Object|Array|undefined}
```

row: 로우 인덱스 <필수> : [number]

```
31 var rowdata = QCell1.getRowData(3);
```

See result

m-0004.html 실행 후 개발자 도구에서 실행한 결과

```
> qcell.getRowData(2)
< ▶ {check1: true, check2: "ng", check3: "1", isActive: false, balance: "3,567,933.28", ...}
>
```

6) setRowData

QCELL 객체에 반영된 데이터 중 지정한 행의 데이터를 갱신하는 API 입니다.

Syntax & eXample

```
setRowData(row, data) → {boolean}
```

row: 로우 인덱스 <필수> : [number]

data: 컬럼 단위 데이터 <필수> : [Object | Array]

```
39 //QCELL의 data 타입이 "object"인 경우
40 QCell1.setRowData(3, {"name" : "kevin", "age" : "35"});
41
42 //QCELL의 data 타입이 "array"인 경우
43 QCell1.setRowData(4, ["kevin", "35"]);
```

See result

m-0004.html 실행 후 개발자 도구에서 아래 명령(firstname, lastname 을 변경)을 실행한 결과

```
> qcell.setRowData(1, {'firstname': 'James', 'lastname': 'Kwak'});
< true
> |
```

setData 확인하기

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	st
1	<input type="checkbox"/>	James	Kwak						
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech F
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech F

7) setRowDataEx

QCELL 은 입력받은 행 인덱스의 "행 단위 데이터"와 숨겨진 행 인덱스"행 단위 데이터"까지 갱신합니다.

```
setRowDataEx(row, data) → {boolean}
```

row: 행 인덱스 <필수> : [number]

data: 행 단위 데이터 <필수> : [Object | Array]

//QCELL 의 data 타입이 "object"인 경우

```
qcell.setRowDataEx(3, { "name": "kevin", "age": "35" });
```

//QCELL 의 data 타입이 "array"인 경우

```
qcell.setRowDataEx(4, ["kevin", "35"]);
```

8) getFocusRowData

현재 포커스된 로우의 데이터를 반환합니다.

 Syntax & eXample

```
getFocusRowData() → {boolean}
```

```
29 QCell11.getFocusRowData();
```

 See result

m-0004.html 실행 후 개발자 도구에서 실행한 결과

```
> qcell.getFocusRowData()
< ▶ {check1: true, check2: "ng", check3: "1", isActive: false, balance: "3,567,933.28", ...}
> |
```

9) findRow

입력한 row에서 출발하여 하위 row들의 대상 컬럼에서 입력한 데이터를 찾아서 처음 발견된 row를 반환합니다.

 Syntax & eXample

```
findRow(data, startRow, targetCol) → {number}
```

data: 찾을 데이터 <필수> : [string]
startRow: row 인덱스 <필수> : [number]
targetCol: 컬럼 인덱스 <필수> : [number]

```
30 QCell11.findRow('abcd', 1, 1);
```

 See result

m-0004.html 실행 후 개발자 도구에서 아래 코드를 입력한 결과입니다.

```
> qcell.findRow("Conrad", 1, 3)
< 2
> |
```

10) findRowEx

QCELL 은 해당 row 이하에서 대상 컬럼 중 특정 데이터를 찾습니다.

 Syntax & eXample

```
findRowEx(data, startRow, key) → {number}
```

data : 찾을 데이터 <필수> : [string]
startRow : 찾기 시작 할 행 인덱스 <필수> : [number]
key : 셀의 키 인덱스 <필수> : [string]

```
qcell.findRowEx('Tom', 1, 'firstname')
```

 See result

m-0004.html 실행 후 개발자 도구에서 아래 코드를 입력하면 Tom 이라는 특정 데이터가 몇 행에 있는지 알 수 있습니다.

	<input type="checkbox"/>	firstname↕	lastname↕	age
1	<input type="checkbox"/>	Bobbie	Goodman	44
2	<input type="checkbox"/>	Mclaughlin	Conrad	49
3	<input type="checkbox"/>	Donald	Trump	30
4	<input type="checkbox"/>	Tom	ames	39
5	<input type="checkbox"/>	Mclaughlin	Conrad	41
6	<input type="checkbox"/>	Mclaughlin	Conrad	59

```
> qcell.findRowEx('Tom', 1, 'firstname')
< 4
```

11) swapData

입력한 로우 간 데이터를 치환하는 API 입니다. 이렇게 하면 로우의 데이터를 이동시키는 효과를 얻을 수 있습니다.

Syntax & eXample

```
swapData(srcRow, dstRow) → {boolean}
```

srcRow: 치환하기 위한 로우 인덱스 <필수> : [number]

dstRow: 치환하기 위한 대상 로우 인덱스 <필수> : [number]

```
30 QCell11.swapData( 1, 5);
```

See result

m-0004.html 실행 후 개발자 도구에서 실행한 결과(실행 전, 후 비교하여 확인)

```
> qcell.swapData(1,2)
< true
> |
```

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Av
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
2	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Av

12) sort

sort 옵션이 true 인 컬럼에 대하여 API 를 통해 정렬을 시킬 수 있습니다.

 Syntax & eXample

`sort(sortlist) → {boolean}`

sortlist: 정렬 조건 <필수>

sortlist[i].col: 컬럼 인덱스: **[number]**

sortlist[i].sorttype: 정렬 타입 ("asc" | "desc"): **[string]**

sortlist[i].datatype: 데이터 타입 ("string" | "number"): **[string]**

```
30 //3번째 열을 기준으로 (string)오름차순으로 정렬 (단일 정렬)
31 QCell1.sort([[3, "asc", "string"]]);
32 //3번째 열을 기준으로 (string)오름차순으로, 1번째 열을 기준으로 (number)내림차순으로 정렬 (다중 정렬)
33 QCell1.sort([[3, "asc", "string"],[1, "desc", "number"]]);
```

 See result

m-0004.html 실행 후 개발자 도구에서 실행한 결과(age 컬럼을 대상으로 정렬 전과 후 확인)

```
> qcell.sort([[4, "asc", "number"]])
< true
> |
```

	<input type="checkbox"/>	firstname↕	lastname↕	age ↕	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Av
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
3	<input type="checkbox"/>	Donald	Trump	30	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
4	<input type="checkbox"/>	Tom	James	39	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

	<input type="checkbox"/>	firstname↕	lastname↕	age ↕	phone	birthday	company	city	street
1	<input type="checkbox"/>	Mclaughlin	Conrad	29	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
2	<input type="checkbox"/>	Donald	Trump	30	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
3	<input type="checkbox"/>	Tom	James	39	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
4	<input type="checkbox"/>	Mclaughlin	Conrad	41	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

13) getColData

QCELL 객체에서 지정한 컬럼의 데이터를 배열로 반환합니다.

 Syntax & eXample

```
getColData(col) → {Array}
```

col: 컬럼 인덱스 <필수> : [number]

```
39 QCell1.getColData(3);
```

 See result

m-0004.html 실행 후 개발자 도구에서 명령을 실행한 결과

```
> qcell.getColData(4)
< ▶ (13) [44, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49, 49]
> |
```

14) setColData

QCELL 객체에 지정한 데이터로 해당 컬럼을 갱신합니다.

 Syntax & eXample

```
setColData(col, data) → {boolean}
```

col: 컬럼 인덱스 <필수> : [number]

data: 컬럼 단위 데이터 <필수> : [Array]

```
30 QCell1.setColData(4, ["kevin", "james"]);
```

 See result

m-0004.html 실행 후 개발자 도구에서 명령을 실행한 결과

```
> qcell.setColData(3, ['곽', '박', '이', '최', '김', '류', '문']);
< true
>
```

setData 확인하기

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	곽	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	박	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
3	<input type="checkbox"/>	Mclaughlin	이	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
4	<input type="checkbox"/>	Mclaughlin	최	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
5	<input type="checkbox"/>	Mclaughlin	김	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
6	<input type="checkbox"/>	Mclaughlin	류	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

15) getCellData

QCELL 객체에서 지정한 셀(로우, 컬럼으로 인덱싱)의 데이터를 반환합니다.

Syntax & eXample

```
getCellData(row, col) → {*}
```

row: 셀의 로우 인덱스 <필수> : [number]

col: 셀의 컬럼 인덱스 <필수> : [number]

See result

m-0004.html 실행 후 개발자 도구에서 아래 코드를 입력한 결과입니다.

```
> qcell.getCellData(1,3)
< "Goodman"
> |
```

setData 확인하기

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	str
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech PI
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech PI

 **check eXample** 전체 소스는 manual 내 "QCELL의 데이터 변경하기(setData API 동작 확인)" 참조

16) setCellData

QCELL 객체의 지정한 셀(로우, 컬럼)에 대한 데이터를 갱신합니다.

Syntax & eXample

```
setCellData(row, col, data) → {boolean}
```

row: 셀의 로우 인덱스 <필수> : [number]

col: 셀의 컬럼 인덱스 <필수> : [number]

data: 셀 데이터 <필수> : [*]

```
31 QCell1.setCellData(3, 1, "test");
```

See result

m-0004.html 실행 후 개발자 도구에서 아래 코드를 입력한 결과입니다.

```
> qcell.setCellData(1,3, '곽')
< true
> |
```

setData 확인하기

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	str
1	<input type="checkbox"/>	Bobbie	곽	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech PI
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech PI

17) getCellLabel

QCELL 은 selectmenu(combo list) 컬럼 타입을 지원합니다. 이 타입의 데이터는 label 과 value 로 구성되어 있습니다. 예를 들어 지역이라면 보이는 것은 '서울'이지만 값을 처리할 때는 '02'로 할 수 있습니다. 하나 더 예를 들면 성별 컬럼에 보이는 것은 '남성'이지만 값은 'm'으로 처리할 수 있으며, 이 때 지정한 셀의 현재 표기된 데이터를 반환합니다.

checkbox, image, html, datepicker 타입의 컬럼도 실제 내용과 보여지는 것이 다르므로 이러한 컬럼에서 사용합니다.

Syntax & eXample

```
getCellLabel(row, col) → {*}
```

row: 데이터 반환할 셀의 로우 인덱스 <필수> : [number]

col: 데이터 반환할 셀의 컬럼 인덱스 <필수> : [number]

```
30 QCell1.getCellLabel(3, 1);
```

See result

```
> qcell.getCellLabel(2,4)
< "미사용"
> qcell.getCellData(2,4)
< "N"
> |
```

<input type="checkbox"/>	카테고리ID ↕	등록날짜	카테고리명 ↕	사용여부	설명 ↔	등록자 ↕
<input type="checkbox"/>	FOOD-00001	2019-04-...	라면류	사용	라면제품에 대한 카테고리입니다.	홍길동
<input checked="" type="checkbox"/>	FOOD-00002	2019-04-...	주류	미사용	주류에 대한 카테고리입니다.	김길동
<input type="checkbox"/>	FOOD-00003	2019-04-...	우유류	사용	우유류와 유제품에 대한 카테고리입니다.	박길동
<input type="checkbox"/>	FOOD-00004	2019-04-...	통조림류	사용	캔으로된 통조림에 대한 카테고리입니다.	최길동

check eXample 전체 소스는 manual 내 "selectmenu(combo list) 데이터 확인" 참조

18) setCellLabel

QCELL 객체의 값이 아닌 눈에 보이는 label 값을 변경합니다.

Syntax & eXample

```
setCellLabel(row, col, data) → {boolean}
```

row: 셀의 로우 인덱스 <필수> : [number]

col: 셀의 컬럼 인덱스 <필수> : [number]

data: 셀 데이터 <필수> : [*]

```
32 QCell1.setCellLabel(3, 1, '출력');
```

See result

m-0005 html 파일을 실행 후 개발자 도구를 열어 아래 코드를 입력해주세요.

```
> qcell.setCellLabel(2, 4, '사용');
< true
>
```

<input type="checkbox"/>	카테고리ID	등록날짜	카테고리명	사용여부	설명	등록자
<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	미사용	라면제품에 대한 카테고리입니다.	홍길동
<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	미사용	주류에 대한 카테고리입니다.	김길동
<input type="checkbox"/>	FOOD-00003	2019-04-10	우유류	사용	우유류와 유제품에 대한 카테고리입니다.	박길동
<input type="checkbox"/>	FOOD-00004	2019-04-10	통조림류	사용	캔으로된 통조림에 대한 카테고리입니다.	최길동

화면에 표시된 "미사용" label 값을 "사용" label 값으로 변경합니다.

<input type="checkbox"/>	카테고리ID	등록날짜	카테고리명	사용여부	설명	등록자
<input type="checkbox"/>	FOOD-00001	2019-04-...	라면류	사용	라면제품에 대한 카테고리입니다.	홍길동
<input checked="" type="checkbox"/>	FOOD-00002	2019-04-...	주류	사용	주류에 대한 카테고리입니다.	김길동
<input type="checkbox"/>	FOOD-00003	2019-04-...	우유류	사용	우유류와 유제품에 대한 카테고리입니다.	박길동

More info. 화면에 표시된 label 값을 변경하며 실제 객체값은 변경하지 않습니다.

19) getCellDataEx

QCELL 객체에 바인딩된 데이터가 모두 화면의 컬럼에 나오지 않을 경우 다른 Key 들은 숨은 컬럼 형태로 존재하게 됩니다. 이 컬럼의 특정 셀의 데이터를 얻어서 사용할 때 해당 로우와 Key 명을 주고 반환합니다.

Syntax & eXample

```
getCellDataEx(row, key) → {*}
```

row: 셀의 로우 인덱스 <필수> : [number]

key: 셀의 Key <필수> : [string]

```
30 QCell1.getCellDataEx(3, 'name');
```

 See result

m-0005.html 실행 후 개발자 도구에서 아래 코드를 입력해주세요.

2000, 3000 값은 price 라는 Key 의 값들로 화면에 컬럼으로 바인딩 안되어 있는 상태지만 getCellDataEx API 를 사용하면 해당 값들을 반환할 수 있습니다.

```
> qcell.getCellDataEx(1, 'price');
< 2000
> qcell.getCellDataEx(2, 'price');
< 5000
>
```

20) setCellDataEx

getCellDataEx 와 반대로 숨은 컬럼(Key)에 바인딩된 데이터를 화면상에서는 수정할 수 없으므로 이 API 를 사용해 수정합니다.

 Syntax & eXample

```
setCellDataEx(row, key, data) → {boolean}
```

row: 셀의 로우 인덱스 <필수> : [number]

key: 셀의 Key <필수> : [string]

data: 셀 데이터 <필수> : [*]

```
30 QCell1.getCellDataEx(3, 'name');
```

 See result

숨긴 컬럼(price)의 1 번째 로우의 값을 변경 후 getCellDataEx API 를 사용하여 확인해주세요.

```
> qcell.setCellDataEx(1, 'price', 10000);
< true
> qcell.getCellDataEx(1, 'price');
< 10000
> |
```

21) getInitCellDataEx

QCELL 은 행, key 에 해당하는 셀의 초기화 데이터를 반환합니다.

JS Syntax & eXample

```
getInitCellDataEx(row, key) → {*}
```

row: 셀의 행 인덱스 <필수> : [number]

col: 셀의 key <필수> : [string]

```
qcell.getInitCellDataEx(3, 'name');
```

See result

manual 폴더의 m-0028.html 파일을 실행해주세요.

- ① 이름 컬럼 1 행의 고길동 데이터를 홍길동으로 변경합니다.

이름	전화번호	이름	전화번호	이메일
고길동	010-1234-5678	홍길동	010-1234-5678	gogildong@gamil.com
둘리	010-2345-6789	둘리	010-2345-6789	twori@gamil.com
마이콜	010-3456-7890	마이콜	010-3456-7890	lamen@gamil.com
도우너	010-1111-2222	도우너	010-1111-2222	dunkin@gamil.com
또치	010-2222-3456	또치	010-2222-3456	again@gamil.com
박희동	010-1234-5678	박희동	010-1234-5678	hidong@gamil.com
고길동	010-1234-5678	고길동	010-1234-5678	gogildong@gamil.com
둘리	010-2345-6789	둘리	010-2345-6789	twori@gamil.com
마이콜	010-3456-7890	마이콜	010-3456-7890	lamen@gamil.com
도우너	010-1111-2222	도우너	010-1111-2222	dunkin@gamil.com

고길동 => 홍길동으로 변경

- ② getInitCellDataEx 를 사용하면 변경된 값이 아닌 초기 값을 가져올 수 있습니다.

4.2. 추가/수정/삭제

1) addRow

QCELL 에 로우를 추가하는 방법으로 마지막 로우 다음에 추가하는 API 입니다. 빈 로우를 추가만 할 수도 있으며, 추가하면서 파라미터로 해당 로우에 데이터를 함께 넣을 수도 있습니다.

Syntax & eXample

```
addRow(dataopt) → {boolean}
```

data: 로우 데이터 <선택> : [Object] 또는 [Array]

```
30 QCell1.addRow();
31 QCell1.addRow({"name" : "kevin", "age" : "35"}); //QCELL의 data 타입이 "object"인 경우
32 QCell1.addRow(["kevin", "35"]); //QCELL의 data 타입이 "array"인 경우
```

See result

m-0006.html 샘플에서 “마지막 로우 다음에 로우 추가” 버튼을 클릭하면 아래 코드가 실행되고 그 결과는 아래 화면과 같습니다.

```
34 qcell.addRow({'firstname':'노건', 'lastname':'곽', 'age':50, 'phone':'010-1234-5678', 'birthda
```

마지막 로우 다음에 로우 추가		2번째 로우에 추가(insertRow)							
	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
4	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
5	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
6	<input type="checkbox"/>	노건	곽	50	010-1234-5...	1990/01/21	라잇텍		

 **check eXample** 전체 소스는 manual 내 "로우 추가(addRow,insertRow)에 대한 API 확인" 참조

2) addRows

addRow 의 확장으로 QCELL 은 마지막 로우 다음에 한 로우가 아닌 여러 로우를 추가할 수 있고 로우 수가 아닌 추가될 데이터를 넣을 수도 있습니다. 이 경우 데이터의 개수만큼 로우가 자동으로 추가됩니다.

Syntax & eXample

`addRows(count_or_data) → {boolean}`

count: 로우의 개수 <선택> : [number]

data: 데이터 배열 <선택> : [Array] 또는 [Object]

```
30 QCell1.addRow(3);
31 //QCELL의 data 타입이 "object"인 경우
32 QCell1.addRows([{"name" : "kevin", "age" : "35"}, {"name" : "jenny", "age" : "28"}]);
33 QCell1.addRows([["kevin", "35"], ["jenny", "28"]]); //QCELL의 data 타입이 "array"인 경우
```

See result

m-0006.html 샘플을 실행 후 개발자 도구에서 아래 코드를 입력한 결과 화면입니다.

```
> qcell.addRow(3)
< true
> |
```

5	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Pla
6	<input type="checkbox"/>								
7	<input type="checkbox"/>								
8	<input type="checkbox"/>								

3) insertRow

지정한 로우 인덱스 위치에 로우를 추가하는 API 입니다.

이 때 데이터도 함께 파라미터로 제공하면 로우 추가와 함께 데이터가 추가되는 로우에 반영이 됩니다. 또한, addRow 와 동일한 동작이며 추가되는 로우에 대한 인덱스를 지정할 수 있습니다.

Syntax & eXample

```
insertRow(row, data_opt) → {boolean}
```

row: 로우 인덱스 <필수> : [number]

data: 로우 데이터 <선택> : [Object] 또는 [Array]

```
30 QCell1.insertRow(3);
31 QCell1.insertRow(3, {"name" : "kevin", "age" : "35"}); //QCELL의 data 타입이 "object"인 경우
32 QCell1.insertRow(3, ["kevin", "35"]); //QCELL의 data 타입이 "array"인 경우
```

See result

m-0006.html 샘플에서 "2 번째 로우에 추가(insertRow)" 버튼을 클릭 후 아래 코드를 입력하면 아래와 같은 결과를 볼 수 있습니다.

```
30 qcell.insertRow(2,{'firstname':'노건', 'lastname':'곽', 'age':50, 'phone':'010-1234-5678', 'b
```

마지막 로우 다음에 로우 추가		2번째 로우에 추가(insertRow)							
	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	노건	곽	50	010-1234-5...	1990/01/21	라잇텍		
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
4	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
5	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
6	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

 **check eXample** 전체 소스는 manual 내 "로우 추가(addRow,insertRow)에 대한 API 확인" 참조

4) insertRows

addRows 와 동일하게 여러 로우를 추가하는데 특정 로우 인덱스에 추가하는 API 입니다. 추가 시 몇 개의 로우를 추가할지 count 를 입력하거나 추가될 로우들의 데이터를 입력하면 자동으로 해당 데이터 수만큼 로우가 생성되고 데이터가 반영됩니다.

 Syntax & eXample

`insertRows(row, count_or_data) → {boolean}`

- row: 로우 인덱스 <필수> : [number]
- count: 로우의 개수 <필수> : [number]
- data: 로우 데이터 배열 <필수> : [Array]

```
30 QCell1.insertRows(3, 2);
31 //QCELL의 data 타입이 "object"인 경우
32 QCell1.insertRows(3, [{"name" : "kevin", "age" : "35"}, {"name" : "jenny", "age" : "28"}]);
33 //QCELL의 data 타입이 "array"인 경우
34 QCell1.insertRows(3, [{"kevin", "35"}, ["jenny", "28"}]);
```

 See result

m-0006.html 실행 후 개발자 도구에서 아래 코드를 입력하면 그 결과는 아래 화면과 같습니다.

```
> qcell.insertRows(3, 2)
< true
> |
```

2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Pla
3	<input type="checkbox"/>								
4	<input type="checkbox"/>								
5	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Pla

 **More info.** `insertRowsEx` API 는 `insertRow` 를 여러 인덱스에서 실행할 때 하나의 API 에 추가될 위치와 데이터를 여러 개 넣어 한 번에 처리할 수 있도록 해 줍니다.

5) `removeRow`

QCELL 객체 화면의 로우 중 마지막 로우를 삭제하는 API 입니다. 마지막 로우 하나를 삭제하는 동작이므로 입력 값은 없습니다.

 Syntax & eXample

`removeRow() → {boolean}`

```
30 QCell1.removeRow();
```

 See result

m-0006.html 실행 후 개발자 도구에서 아래 코드를 입력하면 그 결과는 아래 화면과 같습니다.

```
> qcell.removeRow()
< true
> |
```

코드 실행 전

4	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
5	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

코드 실행 후

4	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

6) removeRows

QCELL 객체에서 마지막 로우부터 입력한 수만큼 로우를 삭제합니다.

 Syntax & eXample

```
removeRows(count) → {boolean}
```

count: 삭제할 로우의 개수 <필수> : [number]

```
QCell11.removeRows(3);
```

 See result

m-0006.html 샘플을 실행 후 개발자 도구에서 아래 코드를 입력하면 그 결과는 아래 화면과 같다.

```
> qcell.removeRows(3)
< true
> |
```

코드 실행 전

2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
4	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
5	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

코드 실행 후

2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
---	--------------------------	------------	--------	----	--------------	------------	----------	----------	------------

7) deleteRow

removeRow 와 달리 QCELL 객체에서 입력한 로우 인덱스에 있는 로우를 삭제하는 것으로 화면에서 지워지는 것뿐만 아니라 실제 데이터에서도 해당 로우의 데이터가 삭제되는 API 입니다. 비워지는 방식이 아닌 삭제된 로우 아래의 데이터들이 위로 채워지는 방식으로 동작합니다.

Syntax & eXample

```
deleteRow(row) → {boolean}
```

row: 로우 인덱스 <필수> : [number]

```
30 QCell1.deleteRow(3);
```

See result

m-0006.html 실행 후 개발자 도구에서 아래 코드를 입력하면 그 결과는 아래 화면과 같습니다.

```
> qcell.deleteRow(1)
< true
> |
```

코드 실행 전

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

코드 실행 후

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

8) deleteRows

QCELL 객체에서 입력한 특정 row 인덱스부터 입력한 수만큼 row를 지웁니다. 또한, 화면뿐만 아닌 데이터도 함께 삭제가 됩니다. 지워진 자리는 아래 row들이 위로 채워지는 방식입니다.

 Syntax & eXample

```
deleteRows(row, count) → {boolean}
```

row: row 인덱스 <필수> : [number]

count: 삭제할 row의 개수 <필수> : [number]

```
30 QCell11.deleteRows(3, 2);
```

 See result

m-0006.html 샘플을 실행 후 개발자 도구에서 아래 코드를 입력하면 그 결과는 아래 화면과 같다.

```
> qcell.deleteRows(2,3)
< true
> |
```

코드 입력 전

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
4	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
5	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

코드 실행 후

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

 More info. **removeRowsEx** API 는 deleteRow 를 여러 인덱스에서 실행할 때 하나의 API 에 삭제될 위치를 여러 개 나열하여 입력 후 한 번에 처리할 수 있도록 해줍니다.

9) insertStaticRow

입력받은 행 인덱스 위치에 고정출력 행을 삽입합니다.

Syntax & eXample

```
insertStaticRow(row, dataopt) → {boolean}
```

row : 행 인덱스 <필수> : [number]

data : <optional> 행 데이터 <필수> : [object][Default :{ }]

See result

manual 폴더안에 m-0004.html 파일을 열어 개발자 도구를 띄운 후 아래 코드를 입력해주세요

```
qcell.insertStaticRow(1, {'firstname': '민정', 'lastname': '이', 'age': '28', 'phone': '010-1234-5678', 'birthday': '1994/05/17', 'company': '라잇텍', 'city': '서울' })
```

firstname	lastname	age
민정	이	28
Bobbie	Goodman	44
Mclaughlin	Conrad	49
Donald	Trump	30
Tom	James	39
Mclaughlin	Conrad	41

```
> qcell.insertStaticRow(1,{'sequence':'checkbox','firstname':'민정','lastname':'이','age':5678,'birthday':'1994/05/17','company':'라잇텍','city':'서울'})
< true
>
```

 **check eXample** 전체 소스는 manual 내 "상태값이 변경된 로우 정보(getState, getStates)를 반환하는 API 확인"

10) deleteStaticRow

입력받은 행 인덱스 위치의 고정출력 행을 삭제합니다. (일반적인 행 삭제는 고정출력 행 정보가 삭제되지 않아, 반드시 고정출력 행은 해당 API 를 거쳐 삭제해야 합니다.)

Syntax & eXample

```
deleteStaticRow(row) → {boolean}
```

nCol : 행 인덱스 <필수> : [number]

앞서 추가한 고정 출력행을 deleteStaticRow API 를 통해 삭제할 수 있습니다.

firstname↕	lastname↕	age ↕	phone
Bobbie	Goodman	44	870-4412-705
McLaughlin	Conrad	49	938-4902-720
Donald	Trump	30	938-4902-720
Tom	James	39	938-4902-720

```
> qcell.deleteStaticRow(1)
▶ 9 [Violation]Added non-passive event listener to a scroll event on page more responsive. See <URL>
< true
>
```

11) isStaticRow

QCELL 은 특정 행이 출력전용 행인 static row 인지 확인합니다.

Syntax & eXample

```
isStaticRow(nCol) → {Number}
```

nCol : 행 인덱스 <필수> : [number]

```
qcell.isStaticRow(1);
```

12) getState

QCELL 객체는 필요에 따라 로우가 추가 및 삭제, 데이터가 변경되기도 합니다. 이 때 로우의 상태 값이 있어서 변경된 로우를 얻어서 처리할 때 사용하는 API 입니다. 상태는 d(삭제), i(추가), u(변경, 수정)가 있으면 이 상태 값이 아닌 로우들은 상태 값이 변경되지 않은 것입니다.

Syntax & eXample

```
getState(state) → {Array}
```

state: 반환할 로우 상태 정보의 타입("i" | "u" | "d") <필수> : [string]

```
30 QCell11.getState('i');
```

Do programming

삭제된 로우의 수를 얻어내어 화면에 정보를 뿌려주는 코드

```
32     if (qcell.getState('d').length > 0)
33         message = '삭제된 로우 개수: ' + qcell.getState('d').length;
34     else message = '삭제된 로우가 없습니다.'
35     writeMessage(message);
```

 See result

m-0007.html 샘플은 m-0006 을 확장한 것으로 실행 후 로우 삭제나 추가 전에 “삭제된 로우 개수 확인” 버튼을 클릭해 상태 값을 체크

값이 보여질 영역:
삭제된 로우가 없습니다.

이제 “4 번째 로우 삭제” 버튼을 눌러 QCELL 객체의 값을 변경시킨 후 위와 동일하게 변경된 상태의 값을 확인

값이 보여질 영역:
삭제된 로우 개수: 1

 **check eXample** 전체 소스는 manual 내 “상태값이 변경된 로우 정보(getState, getStates)를 반환하는 API 확인” 참조

13) clearStates

삭제정보를 포함한 모든 상태정보를 삭제하는 API 입니다.

 Syntax & eXample

`clearStates()` → {boolean}

QCell1.clearStates

14) getStates

getState API 는 특정 상태 값을 얻는데 반하여 getStates 는 모든 상태 값을 한 번에 얻어내는 API 입니다. 따라서 입력 값이 필요하지 않습니다.

 Syntax & eXample

`getStates()` → {Object}

```
30     QCell1.getStates();
```

 Do programming

추가된 로우가 있으면 몇 개가 추가되었는지 그리고 그 중에 첫 번째 로우의 firstname 값을 뿌려주는 코드

```
40     if(qcell.getStates().i.length > 0)
41         message = '추가된 로우의 행은 ' + qcell.getStates().i[0].row + '이고 '
42             + 'firstname은 ' + qcell.getStates().i[0].data.firstname + '입니다.'
43     else
44         message = '추가된 로우가 없습니다.';
45     writeMessage(message);
```

 See result

m-0007.html 샘플은 m-0006 을 확장한 것으로 실행 후 “추가된 로우의 fname 확인” 버튼을 클릭해 상태 값을 체크

값이 보여질 영역:
추가된 로우가 없습니다.

이제 “2 번째 로우에 추가” 버튼을 눌러 QCELL 객체의 값을 변경시킨 후 위와 동일하게 변경된 상태의 값을 확인

값이 보여질 영역:
추가된 로우의 행은 2이고 firstname은 노건입니다.

 **check eXample** 전체 소스는 manual 내 “상태값이 변경된 로우 정보(getState, getStates)를 반환하는 API 확인” 참조

 **More info.** **clearStates** API 는 상태가 변경되어 가지고 있는 값을 모두 지워서 초기화 하는 동작을 합니다. **getStatesEx** API 는 상태가 변경된 값들을 상태별이 아닌 object 형태로 반환, 이 때 각 로우별 상태 값을 나타내는 속성을 status 라는 속성의 값으로 표기하는데 status 가 아닌 다른 속성으로 정의하고자 할 경우 **PropertyName** 입력하면 됩니다. **getStatesBySep** API 는 변경된 로우의 값을 object 가 아닌 구분자 방식의 문자열로 반환하는 것으로 일부 서버 프로그램들이 예전의 데이터 통신 방식으로 구분자를 두어 처리하던 것을 지원한다. 모든 데이터를 구분자 방식으로 반환하는 것은 **getAllStatesBySep** API 입니다.

15) getStatesEx

QCELL 은 행 상태 정보들을 반환합니다.

 **Syntax & eXample**

getStatesEx(propertyName) → {Object}

propertyName : (optional)프로퍼티 명칭 <필수> : [string]

```
qcell.getStatesEx()
```

 See result

manual 폴더의 m-0006.html 파일을 열어 실행하여 (마지막 로우 다음에 로우 추가 버튼)을 눌러 아래 코드를 입력해주세요.

```
> qcell.getStatesEx()
< ▼ (4) [{...}, {...}, {...}, {...}] ⓘ
  ▶ 0: {row: 6, data: {...}, status: 'i'}
  ▶ 1: {row: 7, data: {...}, status: 'i'}
  ▶ 2: {row: 8, data: {...}, status: 'i'}
  ▶ 3: {row: 9, data: {...}, status: 'i'}
  length: 4
  ▶ [[Prototype]]: Array(0)
```

	<input type="checkbox"/>	firstname	lastname	age	phone	birthday	company	city	street
4	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
5	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
6	<input type="checkbox"/>	노건	곽	50	010-1234-5678	1990/01/21	라잇텍		
7	<input type="checkbox"/>	노건	곽	50	010-1234-5678	1990/01/21	라잇텍		
8	<input type="checkbox"/>	노건	곽	50	010-1234-5678	1990/01/21	라잇텍		
9	<input type="checkbox"/>	노건	곽	50	010-1234-5678	1990/01/21	라잇텍		

추가한 행의 상태 정보들을 반환해주는 걸 볼 수 있습니다.

16) setRowState

QCELL 은 행 상태정보를 설정합니다.

 Syntax & eXample

```
setRowState(row, state) → {boolean}
```

row : 설정할 행 인덱스 <필수> : [number]

state : 설정할 상태 값 <필수> : [string]

```
qcell.setRowState(3, 'd');
```

 See result

manual 폴더를 열어 m-0030.html 파일을 실행해주세요.

라디오 버튼은 행을 삭제하면 주황색 또는 행을 추가하면 파란색으로 표시되도록 설정하였습니다.

	<input type="checkbox"/>	<input type="radio"/>	custom	카테고리ID	카테고리명	사용여부	설명	등록자
1	<input type="checkbox"/>	<input type="radio"/>	1	FOOD-00001	라면류	Y	라면제품에 대한 카테고리입니다.	홍길동
2	<input type="checkbox"/>	<input type="radio"/>	2	FOOD-00002	주류	N	주류에 대한 카테고리입니다.	김길동
3	<input type="checkbox"/>	<input type="radio"/>	3	FOOD-00003	우유류	Y	우유류와 유제품에 대한 카테고...	박길동
4	<input type="checkbox"/>	<input type="radio"/>	4	FOOD-00004	통조림류	Y	캔으로된 통조림에 대한 카테고...	최길동

	<input type="checkbox"/>	<input type="radio"/>	custom	카테고리ID	
	<input type="checkbox"/>	<input checked="" type="radio"/>	1	FOOD-00001	<pre>> qcell.setRowState(1, 'd') < true</pre>
	<input type="checkbox"/>	<input type="radio"/>	2	FOOD-00002	
	<input type="checkbox"/>	<input type="radio"/>	3	FOOD-00003	
	<input type="checkbox"/>	<input type="radio"/>	4	FOOD-00004	

setRowState 을 API 를 통해 해당 1 행의 상태를 행 삭제한 상태로 강제로 설정하였습니다.

	<input type="checkbox"/>	<input type="radio"/>	custom	카테고리ID	
	<input type="checkbox"/>	<input checked="" type="radio"/>	1	FOOD-00001	<pre>> qcell.setRowState(1, 'i') < true</pre>
	<input type="checkbox"/>	<input type="radio"/>	2	FOOD-00002	
	<input type="checkbox"/>	<input type="radio"/>	3	FOOD-00003	
	<input type="checkbox"/>	<input type="radio"/>	4	FOOD-00004	

setRowState 을 API 를 통해 해당 1 행의 상태를 행 추가한 상태로 강제로 설정하였습니다.

17) getRowState

QCELL 은 행 상태정보를 반환합니다.

 Syntax & eXample

```
getRowState(row) → {string}
```

row : 반환할 행 인덱스 <필수> : [number]

```
qcell.getRowState(1);
```

 See result

앞서 m-0030.html 실행하여 setRowState 를 통해 설정한 값들을 getRowState 를 통해 상태정보의 반환값을 살펴볼 수 있습니다.

```
> qcell.getRowState(1)
< 'd'
```

```
> qcell.getRowState(1)
< 'i'
```

18) removeRowState

QCELL 은 행 상태정보를 삭제합니다.

 Syntax & eXample

```
removeRowState(row) → {boolean}
```

row : 설정할 행 인덱스 <필수> : [number]

```
qcell.removeRowState(1, 'd');
```

19) clearRowStates

QCELL 은 모든 상태정보를 삭제합니다.

 Syntax & eXample

```
clearRowStates() → {boolean}
```

```
qcell.clearRowStates();
```

20) getFocusRowState

QCELL 은 포커스된 행의 상태 정보를 반환합니다.

반환할 행 상태 정보의 타입 ("i" | "u" | "d")

Syntax & eXample

`getFocusRowState()` → {Array}

```
qcell.getFocusRowState();
```

21) getFocusRowData

QCELL 은 현재 포커스된 행의 데이터를 반환합니다.

Syntax & eXample

`getFocusRowData()` → {boolean}

```
qcell.getFocusRowData();
```

manual 폴더의 m-0030.html 실행하여 1 행을 포커스 한 후 개발자 도구 콘솔창에서 `qcell.getFocusRowData(1)` 코드를 입력하면 아래와 같은 결과를 볼 수 있습니다.

	<input type="checkbox"/>	<input type="radio"/>	custom	카테고리ID	카테고리명	사용여부	설명	등록자
1	<input type="checkbox"/>	<input checked="" type="radio"/>	1	FOOD-00001	라면류	Y	라면제품에 대한 카테고리입니다.	홍길동
2	<input type="checkbox"/>	<input type="radio"/>	2	FOOD-00002	주류	N	주류에 대한 카테고리입니다.	김길동
3	<input type="checkbox"/>	<input type="radio"/>	3	FOOD-00003	우유류	Y	우유류와 유제품에 대한 카테고리...	박길동
4	<input type="checkbox"/>	<input type="radio"/>	4	FOOD-00004	통조림류	Y	캔으로된 통조림에 대한 카테고리...	최길동

```
> qcell.getFocusRowData()
< {id: 'FOOD-00001', name: '라면류', description: '라면제품에 대한 카테고리입니다.', useYn: 'Y', regUser: '홍길동'}
  description: "라면제품에 대한 카테고리입니다."
  id: "FOOD-00001"
  name: "라면류"
  regUser: "홍길동"
  useYn: "Y"
  ▶ [[Prototype]]: Object
```

22) getSelectRowState

QCELL 은 특정 행의 상태 정보를 반환합니다.

반환할 행 상태 정보의 타입 ("i" | "u" | "d")

 Syntax & eXample

```
getSelectRowState(row) → {Array}
```

row : 상태 정보를 확인할 행의 인덱스 <필수> : [number]

```
qcell.getSelectRowState(1);
```

 See result

앞서 m-0030.html 실행하여 1 행에 대해 추가와 삭제 버튼을 사용하여 아래와 같은 상태정보를 알 수 있습니다.

<input type="checkbox"/>		custom	카테고리
<input type="checkbox"/>	<input checked="" type="radio"/>	1	
<input checked="" type="checkbox"/>	<input type="radio"/>	2	FOOD-
<input type="checkbox"/>	<input type="radio"/>	3	FOOD-

```
> qcell.getSelectRowState(1)
< ▼ ['i'] ⓘ
  0: "i"
  length: 1
  ▶ [[Prototype]]: Array(0)
```

<input type="checkbox"/>		custom	카테고리ID
<input type="checkbox"/>	<input checked="" type="radio"/>	1	FOOD-00001
<input type="checkbox"/>	<input type="radio"/>	2	FOOD-00002
<input type="checkbox"/>	<input type="radio"/>	3	FOOD-00003
<input type="checkbox"/>	<input type="radio"/>	4	FOOD-00004

```
> qcell.getSelectRowState(1)
< ▶ ['d']
>
```

23) getStatesBySep

QCELL 은 행 상태 정보들을 구분자 형태로 반환합니다.

 Syntax & eXample

```
getStatesBySep(rowSep, colSep) → {string}
```

rowSep : (optional) 행 구분자 <필수> : [string]

colSep : 상태 정보를 확인할 행의 인덱스 <필수> : [number]

```
qcell.getStatesBySep();
```

m-0030.html 실행하여 삭제 버튼을 누른 후 아래 코드를 입력하면 삭제된 행의 상태정보가 구분자 형태로 반환되는 것을 볼 수 있습니다.

```
> qcell.getStatesBySep();
< 'm^rownum^isChecked^tmHeader^id^name^useYn^description^regUser|d^1^true^D^FOOD-0000
1^라면류^Y^라면제품에 대한 카테고리입니다.^홍길동|'
```

24) getAllStatesBySep

QCELL 은 모든 행 상태 정보들은 구분자 형태로 반환합니다.

Syntax & eXample

```
getAllStatesBySep(strStateName, strRowName) → {string}
```

strStateName : (optional) 상태정보 표시명 <필수> : [string]

strRowName : 행 번호 표시명 <필수> : [string]

```
qcell.getAllStatesBySep();
```

큐셀 그리드가 생성된 페이지에 해당 API 를 사용하면 아래와 같이 모든 행의 정보들을 구분자 형태로 반환해주는 걸 볼 수 있습니다.

```
> qcell.getAllStatesBySep("상태정보 표시명", "행 번호 표시명")
< '상태정보 표시명|행 번호 표시명|2|workSelect|code|name|won|food|check|gubun|gubun2|bigoflag|rownum|m|hi|bye|
끝|시작과^|1||0|211101|외상매입금|3|0|1|0|0|테스트수도광열비현장경비||1||1|1|^2||0|211101|외상매입금|3|0|1|0|0|
테스트수도광열비현장경비||2||2|2|^3||0|299101|외상매입금|3|0|1|0|0|테스트수도광열비현장경비||3||3|3|^4||0|299
101|외상매입금|3|0|1|0|0|테스트수도광열비현장경비||4||4|4|^5||0|299101|외상매입금|3|0|1|0|0|테스트수도광열비현장
경비||5||5|5|^6||0|299101|외상매입금|3|0|1|0|0|테스트수도광열비현장경비||6||6|6|^7||0|211101|외상매입금|3|0|
1|0|0|테스트수도광열비현장경비||7||7|7|^8||0|211101|외상매입금|3|0|1|0|0|테스트수도광열비현장경비||8||8|8|^9
|0|211101|외상매입금|3|0|1|0|0|테스트수도광열비현장경비||9||9|9|^10||0|211101|외상매입금|3|0|1|0|0|테스트수도광
열비현장경비||10||10|10|^'
```

4.3. 기타

1) requestData

QCELL 객체는 Server 와 비동기 통신을 지원하고 있고 결과에 대한 처리를 위한 구조를 포함하고 있습니다. 일반적으로 그리드에서 통신을 하지 않고 화면 단위에서 필요한 서버와의 통신 모듈을 개발하는 것이 일반적입니다.

 Syntax & eXample

`requestData(properties) → {*}`

properties: Server 와 통신시 필요 정보 <필수> : [object]

properties.url: Server 와 통신할 URL 주소 <필수> : [string]

properties.param: Server 와 통신시 전달할 Parameter <선택> : [object]

properties.callback: Server 와 통신 이후 실행할 콜백 함수 <선택> : [function]

```

30     var properties =
31     {
32         url: "/data.do",
33         param: {
34             data1: "1",
35             data2: "2"
36         },
37         callback: fnCallback
38     }
39
40     QCell1.requestData(properties);
    
```

2) checkRequired

QCELL 객체 설정 시 required 컬럼으로 설정할 수 있으며 이 컬럼에는 빈 데이터가 있을 수 없습니다. 빈 데이터가 있는지 체크해서 빈 데이터로 있는 셀들의 정보를 반환합니다.

 Syntax & eXample

`checkRequired() → {Array}`

```

28     QCell1.checkRequired();
    
```

 See result

m-0007.html 를 실행 후 개발자 도구에서 아래 코드를 입력해주세요. 빈 데이터로 있는 셀들의 정보를 반환하는 것을 볼 수 있습니다.

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin		49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

```

> qcell.checkRequired()
< ▼[{}]
```

- ▶ 0: {row: 2, col: 3}
 - length: 1
 - __proto__: Array(0)

```

>
    
```

3) showProgress

QCELL 객체 동작 시 필요에 따라 동작 중이라는 Progress 이미지를 화면에 출력할 때 사용하는 API 입니다. hideProgress API 를 실행하기 전까지 계속 반복적으로 동작합니다.

 **Syntax & eXample**

```
showProgress() → {boolean}
```

```
40 QCell1.showProgress();
```

 **See result**

m-0007.html 를 실행 후 개발자 도구에서 아래 코드를 입력하면 Progress 이미지가 나옵니다.

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
4	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
5	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

4) hideProgress

QCELL 객체에 보인 Progress 이미지를 화면에서 숨김 처리하는 API 입니다. 이때 자연스럽게 사라지도록 하기 위하여 사라지는 시간을 설정할 수 있습니다.

 **Syntax & eXample**

```
hideProgress(time) → {boolean}
```

time: Progress 이미지가 숨겨질 때까지의 시간(ms) <선택> : [number]

```
40 QCell1.hideProgress(300);
```

 **See result**

m-0007.html 를 실행 후 개발자 도구에서 아래 코드를 입력하면 Progress 이미지가 나오고 다음 hideProgress 실행하면 사라짐

```
> qcell.showProgress()
< true
> qcell.hideProgress(300)
< true
> |
```

5) editCell

edit 가능한 Cell 이면 Edit 상태로 들어가도록 합니다.

 **Syntax & eXample**

```
editCell(row, col) → {boolean}
```

row : 행 인덱스 <선택> : [number]

col : 열 인덱스 <선택> : [number]

```
qcell.editCell(2,3)
```

 **See result**

m-0007.html 를 실행 후 개발자 도구에서 아래 코드를 입력하면 edit 상태의 컬럼을 볼 수 있습니다.

```
> qcell.editCell(1,2)
< true
```

	<input type="checkbox"/>	firstname↔	lastname	age	phone	birthday	company	city
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock
4	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock
5	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock

6) closeEditMode

edit 상태를 확인하고 edit 상태이면 종료 설정합니다.

 Syntax & eXample

`closeEditMode()` → {boolean}

`qcell.closeEditMode()`

 See result

m-0007.html 를 실행 후 개발자 도구에서 editCell 을 실행한 컬럼에 closeEditMode 를 적용하면 edit 상태가 종료되는 것을 볼 수 있습니다.

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock
4	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock
5	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock

```
> qcell.closeEditMode()
< true
```

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert A
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Pl
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Pl
4	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Pl
5	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Pl

7) divisionSubmit

서버에 정보를 분할 submit 을 통해, 전체 데이터를 로딩합니다.

 Syntax & eXample

`divisionSubmit(url, pageCnt)` → {boolean}

url : 서버와 통신할 URL 주소 <선택> : [string]

pageCnt : 서버와 통신할 때 마다 전달받을 데이터 갯수 <선택> : [number]

8) getUpdateInfo

QCELL 은 갱신된 화면 위치 반환합니다.

Syntax & eXample

```
getUpdateInfo() → {object}
```

```
qcell.getUpdateInfo();
```

4.4. 출력/숨김

1) setCellInvisible

QCELL 은 셀 내의 데이터를 출력/숨김 상태로 설정합니다.

Syntax & eXample

```
setCellInvisible(row, col, bVisible) → {Boolean}
```

row : 출력/숨김 할 행 인덱스 <필수> : [number]

col : 출력/숨김 할 열 인덱스 <필수> : [number]

bVisible : 출력/숨김 여부 <선택> : [boolean]

```
qcell.setCellInvisible(1, 0, true);
```

해당 열의 row 데이터에 true 값을 주어 데이터 숨김

이름	전화번호	이메일	
	010-1111-1...	qwe@gmail.com	삼전
홍길동	010-1111-1...	qwe@gmail.com	삼전
홍길동			
홍길동	010-1111-1...	qwe@gmail.com	삼전
홍길동	010-1111-1...	qwe@gmail.com	삼전

```
> qcell.setCellInvisible(1,0,true);
< true
```

해당 열의 row 데이터에 false 값을 주어 데이터를 표시함

이름	전화번호	이메일	
홍길동	010-1111-1...	qwe@gmail.com	삼전
홍길동	010-1111-1...	qwe@gmail.com	삼전
홍길동	010-1111-1...	qwe@gmail.com	삼전
홍길동	010-1111-1...	qwe@gmail.com	삼전
홍길동	010-1111-1...	qwe@gmail.com	삼전
홍길동	010-1111-1...	qwe@gmail.com	삼전

```
> qcell.setCellInvisible(1,0,false);
< true
```

4.5. 초기화

1) resetData

QCELL 은 속성 original: true 일 때 사용한 가능한 데이터 초기화합니다.

 Syntax & eXample

```
resetData() → {boolean}
```

4.6. 정렬

1) sort

QCELL 은 데이터를 특정 조건을 기준으로 정렬합니다.

(조건이 2 개 이상이면 다중정렬)

 Syntax & eXample

```
sort(sortlist) → {boolean}
```

sortlist : 정렬 조건 <필수> : [Array]

sortlist[i].col : 열 인덱스<필수> : [number]

sortlist[i].sorttype : 정렬 타입 ("asc" | "desc") <필수> : [string]

sortlist[i].datatype : 데이터 타입 ("string" | "number") <필수> : [string]

2009 기준으로 (숫자)오름자순 정렬 업종별 걸림 기준으로 (문자열)오름자순, 통계분류1 걸림 기준으로 (문자열) 내림자순 정렬												
순번	업종별	통계분류1	통계분류2	통계분류3	2007	2008	2009	2010	2011	2012		
1	전체	사업체 수	사업체 수	전체	11716	11977	12755	14281	16144	18926		
2	전체	사업체 수	사업체 수	없음	10714	10425	11600	13115	14834	16784		
3	전체	사업체 수	사업체 수	100㎡ 미만	237	669	207	262	234	557		
4	전체	사업체 수	사업체 수	100㎡~500㎡ 미만	211	349	338	323	410	619		
5	전체	사업체 수	사업체 수	500㎡~1000㎡ 미만	141	121	146	139	156	304		
6	전체	사업체 수	사업체 수	1000㎡ 이상	412	413	465	442	511	662		
7	전체	사업체 수	구성비	없음	91.4	87	90.9	91.8	91.9	88.7		
8	전체	사업체 수	구성비	100㎡ 미만	2	5.6	1.6	1.8	1.4	2.9		
9	전체	사업체 수	구성비	100㎡~500㎡ 미만	1.8	2.9	2.6	2.3	2.5	3.3		
10	전체	사업체 수	구성비	500㎡~1000㎡ 미만	1.2	1	1.1	1	1	1.6		
11	전체	사업체 수	구성비	1000㎡ 이상	3.5	3.4	3.6	3.1	3.2	3.5		
12	전체	사업체 수	평균 편의시설 연면적	소계	349	334	297	307	309	327		
13	전체	증감	사업체 수	전체	0	0	0	1526	0	0		
14	전체	증감	사업체 수	없음	0	0	0	1515	0	0		
15	전체	증감	사업체 수	100㎡ 미만	0	0	0	55	0	0		
16	전체	증감	사업체 수	100㎡~500㎡ 미만	0	0	0	-15	0	0		
17	전체	증감	사업체 수	500㎡~1000㎡ 미만	0	0	0	-7	0	0		
18	전체	증감	사업체 수	1000㎡ 이상	0	0	0	-23	0	0		
19	전체	증감	구성비	없음	0	0	0	0.9	0	-3.2		

2009 기준으로 (숫자) 오름차순 정렬		업종별 컬럼 기준으로 (문자열) 오름차순, 통계분류1 컬럼 기준으로 (문자열) 내림차순 정렬									
순번	업종별	통계분류1	통계분류2	통계분류3	2007	2008	2009	2010	2011	2012	
1	관광숙박업	증감	평균 편의시설 연면적	소계	0	0	-1056	206	0	-4	
2	관광편의시설업	증감	사업체 수	100㎡ 미만	0	0	-405	-14	0	0	
3	카지노업	증감	평균 편의시설 연면적	소계	0	0	-168	60	0	94	
4	국제회의업	증감률	사업체 수	500㎡~1000㎡ 미만	0	0	-100	0	25	-80	
5	관광편의시설업	증감률	사업체 수	100㎡ 미만	0	0	-86.4	-21.9	44	173.6	
6	여행업	증감	사업체 수	100㎡ 미만	0	0	-70	55	0	0	
7	여행업	증감률	사업체 수	100㎡~500㎡ 미만	0	0	-63.8	-58.8	33	-57.3	
8	여행업	증감률	사업체 수	1000㎡ 이상	0	0	-58.3	0	260	5.6	
9	카지노업	증감률	사업체 수	100㎡ 미만	0	0	-50	0	0	0	
10	여행업	증감률	사업체 수	100㎡ 미만	0	0	-47	69.6	-41.8	174.4	
11	국제회의업	증감	평균 편의시설 연면적	소계	0	0	-44	-45	0	13	
12	카지노업	증감률	사업체 수	100㎡~500㎡ 미만	0	0	-33.3	0	33	0	
13	여행업	증감	사업체 수	100㎡~500㎡ 미만	0	0	-30	-10	0	0	
14	관광숙박업	증감률	사업체 수	100㎡ 미만	0	0	-28	55.6	17.9	69.7	
15	국제회의업	증감률	사업체 수	100㎡~500㎡ 미만	0	0	-25	33.3	33	-20	
16	관광편의시설업	증감	구성비	100㎡ 미만	0	0	-20.1	-1	0	2.6	
17	관광숙박업	증감	사업체 수	100㎡~500㎡ 미만	0	0	-19	11	0	0	
18	관광숙박업	증감률	사업체 수	500㎡~1000㎡ 미만	0	0	-14.1	21.9	-10.1	61.3	
19	관광숙박업	증감률	사업체 수	100㎡~500㎡ 미만	0	0	-13	8.7	33	21.3	

5. Selection

5.1. 선택 방법 설정

사용자가 QCELL 객체를 클릭 시 어떤 단위로 선택하고 포커스하여 보여줄 것인지를 설정하는 것을 선택모드라고 하고 아래의 4 가지 선택 동작을 설정하여 생성할 수 있습니다.

- 1) **cells:** 기본값으로 여러 셀을 선택
- 2) **cell:** 하나의 셀만 선택
- 3) **row:** 로우 단위의 단일 선택
- 4) **rows:** 로우 단위의 여러 로우 선택
- 5) **col:** 컬럼 단위로 하나의 단일 컬럼 선택
- 6) **cols:** 컬럼 단위로 여러 개의 컬럼 선택
- 7) **group:** 그룹핑을 적용한 경우 그룹 단위 선택

Syntax & eXample

```
21 selectmode : "cell",
```

See result

m-0007.html 소스에 selectmode 속성 값을 변경 후 저장 후 실행하면 확인

row 모드

	<input type="checkbox"/>	firstname↔	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input checked="" type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

rows 모드

	<input type="checkbox"/>	firstname↔	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input checked="" type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
3	<input checked="" type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

cell 모드

	<input type="checkbox"/>	firstname↔	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

cells 모드

	<input type="checkbox"/>	firstname↔	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

col 모드

	<input type="checkbox"/>	firstname↔	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

cols 모드

	<input type="checkbox"/>	firstname↔	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

5.2. 로우/컬럼 인덱스 반환

1) getSelectedRow

현재 선택된 셀의 로우의 인덱스를 반환하는 API 입니다. 여러 로우나 셀을 선택한 경우 마지막에 선택된 로우의 값을 반환합니다.

Syntax & eXample

```
getSelectedRow() → {number}
```

```
32 var nSelectedRow = QCell1.getSelectedRow();
```

See result

m-0007.html 실행 후 개발자 도구를 띄워 놓은 후에 셀을 클릭 후 도구에서 아래 코드를 입력하여 반환되는 값을 확인

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input checked="" type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

```
> qcell.getSelectedRow()
< 2
>
```

2) getSelectedRows

하나의 셀이 아닌 여러 셀 또는 여러 로우를 선택한 경우 선택된 셀들의 로우 값을 모두 얻어야 하는 경우에 사용하는 API 입니다. 하나 이상의 값이므로 배열로 반환됩니다.

Syntax & eXample

```
getSelectedRows() → {Array}
```

```
32 var arrSelectedRows = QCell1.getSelectedRows();
```

See result

m-0007.html 실행 후 개발자 도구를 띄워 놓은 후에 여러 로우를 선택 후 도구에서 아래 코드를 입력하여 확인해주세요.

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input checked="" type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input checked="" type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

```
> qcell.getSelectedRows()
< ▼ (2) [1, 2] ⓘ
  0: 1
  1: 2
  length: 2
  ▶ __proto__: Array(0)
>
```

3) getSelectedRowData

QCELL 은 현재 선택된 행 인덱스의 "행 단위 데이터"를 객체(또는 배열)로 반환합니다

Syntax & eXample

```
getSelectedRowData() → {Object|Array|undefined}
```

See result

m-0028.html 파일을 열고 개발자 도구에서 아래 코드를 입력해주세요

```
> qcell11.getSelectedRowData()
```

이름	전화번호	이메일	사명	입사일	주소
고길동	010-1234-5678	gogildong@gamil.comm	길동전자	1978-02-03	서울 특별시 송파구
김너	010-2345-6789	twori@gamil.comm	길동전자	1992-07-21	서울 특별시 송파구
마이클	010-3456-7890	lamen@gamil.comm	실용음악학원	1991-02-15	서울 특별시 송파구
도우너	010-1111-2222	dunkin@gamil.comm	길동전자	1992-07-21	서울 특별시 송파구
또치	010-2222-3456	again@gamil.comm	길동전자	1992-07-21	서울 특별시 송파구

```
> qcell11.getSelectedRowData()
< ▼ [0, 1] ⓘ
  ▶ 0: {name: '고길동', phone: '010-1234-5678', email: 'gogildong@gamil.comm', company: '길동전자', regdate: '1978-02-03', ...}
  length: 1
  ▶ [[Prototype]]: Array(0)
>
```

선택된 해당 행의 데이터들을 객체로 반환해주는 걸 볼 수 있습니다.

4) getSelectedCol

선택한 셀의 컬럼 인덱스를 반환하는 API 입니다. 여러 셀을 선택했을 경우 마지막에 선택된 값을 반환합니다.

Syntax & eXample

```
getSelectedCol() → {Number}
```

```
34 var nSelectedCol = QCell1.getSelectedCol();
```

See result

m-0007.html 실행 후 개발자 도구를 띄워 놓은 후에 셀을 선택 후 도구에서 아래 코드를 입력하여 확인해주세요.

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

```
> qcell.getSelectedCol()
< 4
> |
```

5) getSelectedCols

선택한 셀들의 컬럼 인덱스를 반환하는 API 입니다. 여러 셀을 선택했을 경우에 사용하며 배열 값을 반환합니다.

Syntax & eXample

```
getSelectedCols() → {Array}
```

```
34 var arrSelectedCols = QCell1.getSelectedCols();
```

See result

m-0007.html 실행 후 개발자 도구를 띄워 놓은 후에 여러 셀을 선택 후 도구에서 아래 코드를 입력하여 확인해주세요.

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

```
> qcell.getSelectedCols()
< ▼ (2) [3, 4]
  0: 3
  1: 4
  length: 2
  ▶ __proto__: Array(0)
>
```

6. Scroll

사용자가 마우스를 사용하여 스크롤링하는 것이 아니고 로직을 사용하여 특정한 위치로 상하 스크롤링하거나 좌우 스크롤링을 할 때 사용할 API 들을 제공하고 있습니다.

1) scrollTop

입력받은 로우 인덱스 위치로 세로 스크롤바를 위치시키는 API 입니다. 이동할 수 없는 곳으로 요청할 경우 false 값을 반환합니다.

Syntax & eXample

```
scrollTop(row) → {Array}
```

row: 행 인덱스 <필수> : [number]

```
36 QCell1.scrollTop(3);
```

See result

m-0001.html 실행 후 개발자 도구를 띄워 놓은 후에 도구에서 아래 코드를 입력하여 상하 스크로링의 동작 확인

```
> qcell.scrollTop(6)
< undefined
>
```

코드 실행 전

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert A
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Pla
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Pla

코드 실행 후

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
6	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Pla
7	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Pla
8	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Pla

2) scrollLeft

입력받은 컬럼 인덱스로 좌우 스크롤바를 위치시키는 API 입니다. 이동할 수 없는 곳으로 요청할 경우 false 값을 반환합니다.

Syntax & eXample

```
scrollLeft(col) → {Array}
```

col: 컬럼 인덱스 <필수> : [number]

```
36 QCell1.scrollLeft(3);
```

See result

m-0001.html 실행 후 개발자 도구를 띄워 놓은 후에 도구에서 아래 코드를 입력하여 좌우 스크롤링의 동작 확인

```
> qcell.scrollLeft(4)
< undefined
> |
```

코드 실행 전

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert A
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Pla
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Pla

코드 실행 후

	<input type="checkbox"/>	age	phone	birthday	company	city	street	address	regist
1	<input type="checkbox"/>	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Avenue	501 Bergen ...	2014/08
2	<input type="checkbox"/>	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place	873 Kenmor...	2017/10

3) isScroll

좌우, 상하 스크롤바가 보여지고 있는지 확인하는 API 입니다.

JS Syntax & eXample

`isScroll(type) → {boolean}`

type: 스크롤바 타입 <필수> : [string]

Vertical	세로 스크롤바
Horizontal	가로 스크롤바

```
32 QCell1.isScroll("vertical");
```

See result

m-0001.html 실행 후 개발자 도구를 띄워 놓은 후에 도구에서 아래 코드를 입력하여 좌우 스크롤링의 동작 확인

```
> qcell.isScroll('vertical')
< true
> qcell.isScroll('horizontal')
< true
>
```

	<input type="checkbox"/>	age	phone	birthday	company	city	street	address	regist
1	<input type="checkbox"/>	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Avenue	501 Bergen ...	2014/08
2	<input type="checkbox"/>	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place	873 Kenmor...	2017/10
3	<input type="checkbox"/>	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place	873 Kenmor...	2017/10
4	<input type="checkbox"/>	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place	873 Kenmor...	2017/10
5	<input type="checkbox"/>	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place	873 Kenmor...	2017/10
6	<input type="checkbox"/>	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place	873 Kenmor...	2017/10

7. Style

7.1. 높이 설정&반환

QCELL 객체를 생성할 때 프로퍼티를 통해 header 영역과 data 영역의 높이를 설정할 수 있습니다. 별도의 설정이 없을 때에는 기본값으로 설정됩니다. 현재 설정된 값을 확인하는 API 도 제공합니다.

1) rowHeight

헤더 영역과 데이터 영역의 로우 높이를 프로퍼티에 rowheight 속성으로 pixel 단위로 설정합니다.

Syntax & eXample

Property parameters:

Name	Type	Attributes	Description
rowheight	object	<optional>	로우 높이

{ header: [number], data: [number]}

```
26 "rowheight" : {header: 40, data: 40},
```

See result

m-0007.html 에 아래와 같은 프로퍼티 값을 추가 후 저장하여 브라우저를 통해 실행(열기)후 확인

```
26 rowheight: {header: 20, data:40},
```

	<input type="checkbox"/>	firstname	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave

2) getRowHeight

QCELL 객체에 현재 지정한 영역의 헤더와 데이터의 로우 높이를 반환하는 API 입니다.

Syntax & eXample

```
getRowHeight(type) → {number}
```

type: 영역 구분 <필수> : [string]

header	지정한 영역의 헤더 영역
data	지정한 영역의 데이터 영역

```
30 QCell1.getRowHeight("header");
```

 See result

위의 rowheight 속성 실습 후 개발자 도구를 실행하여 아래 명령으로 현재 설정된 로우의 높이를 확인

```
> qcell.getRowHeight('header');
< 20
> qcell.getRowHeight('data');
< 40
> |
```

7.2. 스타일 지정

1) setColStyle

동적으로 필요한 시점에 지정한 컬럼에 "컬럼 단위 스타일"을 설정하는 API 이며, "컬럼 단위 스타일"은 "셀 단위 스타일"보다 우선순위가 낮습니다.

 Syntax & eXample

```
setColStyle(col, style, typeopt) → {boolean}
```

col: 컬럼 인덱스 <필수> : [number]

style: CSS 스타일 <필수> : [Object]

type: 영역 구분 <필수> : [string]

all(default)	지정한 영역의 헤더 영역과 데이터 영역
header	지정한 영역의 헤더 영역
data	지정한 영역의 데이터 영역

```
26 QCell1.setColStyle(3, {"background-color" : "red", "border-color" : "red"});
27 QCell1.setColStyle(3, {"font-weight" : "bold"}, "data");
```

 See result

m-0007.html 실행 후 개발자 도구를 실행하여 아래 코드를 수행하여 확인

```
> qcell.setColStyle(3, {"background-color" : "#dd9955", "border-color" : "red"});
  qcell.setColStyle(3, {"font-weight" : "bold", "font-size" : "9pt"}, "data");
< true
> |
```

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

2) removeColStyle

동적으로 지정한 컬럼에 "컬럼 단위 스타일"을 제거하는 API 입니다.

Syntax & eXample

```
removeColStyle(col, typeopt) → {boolean}
```

col: 컬럼 인덱스 <필수> : [number]

type: 영역 구분 <선택> : [string]

all(default)	지정한 컬럼의 헤더 영역과 데이터 영역
header	지정한 컬럼의 헤더 영역
data	지정한 컬럼의 데이터 영역

```
26 QCell11.removeColStyle(3);
27 QCell11.removeColStyle(3, "header");
```

See result

위의 setColStyle 실행 후 개발자 도구에서 아래 코드를 수행하여 헤더 영역의 스타일이 제거된 것을 확인

```
> qcell.removeColStyle(3, 'header')
< true
>
```

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

3) setRowStyle

동적으로 필요한 시점에 지정한 로우에 "로우 단위 스타일"을 설정하는 API 입니다.

 Syntax & eXample

`setRowStyle(row, style, typeopt) → {boolean}`

row: 로우 인덱스 <필수> : [number]

style: CSS 스타일 <필수> : [Object]

type: 영역 구분 <선택> : [string]

all(default)	지정한 영역의 헤더 영역과 데이터 영역
header	지정한 영역의 헤더 영역
data	지정한 영역의 데이터 영역

```
28 QCell1.setRowStyle(3, {"background-color" : "red", "border-color" : "red"});
29 QCell1.setRowStyle(3, {"font-weight" : "bold", "header"});
```

 See result

m-0007.html 실행 후 개발자 도구에서 아래 코드를 입력하여 확인해주세요

```
> qcell.setRowStyle(2, {"background-color" : "#dd9955", "border-color" : "red"});
qcell.setRowStyle(2, {"font-weight" : "bold", "header"});
< true
> |
```

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input checked="" type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

4) removeRowStyle

QCELL 은 지정한 로우에 "로우 단위 스타일"을 제거하는 API 를 제공합니다.

 Syntax & eXample

`removeRowStyle(row, typeopt) → {boolean}`

row: 로우 인덱스 <필수> : [number]

type: 영역 구분 <선택> : [string]

all(default)	지정한 영역의 헤더 영역과 데이터 영역
--------------	-----------------------

header	지정한 영역의 헤더 영역
data	지정한 영역의 데이터 영역

```
31 QCell1.removeRowStyle(3);
32 QCell1.removeRowStyle(3, "header");
```

 See result

위의 setRowStyle 실행 후 개발자 도구에서 아래 코드를 입력하여 확인해주세요

```
> qcell.removeRowStyle(2);
   qcell.removeRowStyle(2, "header");
< true
> |
```

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

5) getCellStyle

지정한 인덱스의 셀에 설정된 "셀 단위 스타일" 정보를 반환하는 API 입니다. ("컬럼 단위 스타일"이 함께 설정된 경우라도 "셀 단위 스타일" 정보만 반환합니다.)

 Syntax & eXample

```
getCellStyle(row, col) → {Object}
```

row: 로우 인덱스 <필수> : [number]

col: 컬럼 인덱스 <필수> : [number]

```
31 QCell1.getCellStyle(3, 5);
```

 See result

위의 setRowStyle 실행 후 개발자 도구에서 아래 명령으로 동작 확인

```
> qcell.getCellStyle(2,2)
< ▶ {background-color: "#dd9955", border-color: "red"}
> |
```

6) setCellStyle

동적으로 필요한 시점에 지정한 셀에 "셀 단위 스타일"을 설정하는 API 입니다.

 Syntax & eXample

`setCellStyle(row, col, style) → {boolean}`

row: 로우 인덱스 <필수> : [number]

col: 컬럼 인덱스 <필수> : [number]

style: CSS 스타일 <필수> : [Object]

```
31 QCell1.setCellStyle(3, 5, {"background-color" : "red", "border-color" : "red"});
```

 See result

m-0007.html 실행 후 개발자 도구에서 아래 코드를 수행하여 확인

```
> qcell.setCellStyle(2, 3, {"background-color" : "#dd9955", "border-color" : "red"});
< true
> |
```

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

7) setCellStyles

동적으로 필요한 시점에 지정한 범위 내의 셀들에 "셀 단위 스타일"을 설정하는 API 입니다.

 Syntax & eXample

`setCellStyles(firstrow, firstcol, lastrow, lastcol, style) → {boolean}`

QCELL 객체의 ID. setCellStyles(firstrow, firstcol, lastrow, lastcol, style)

firstrow: 설정할 범위의 시작 로우 인덱스 <필수> : [number]

firstcol: 설정할 범위의 시작 컬럼 인덱스 <필수> : [number]

lastrow: 설정할 범위의 마지막 로우 인덱스 <필수> : [number]

lastcol: 설정할 범위의 마지막 컬럼 인덱스 <필수> : [number]

style : CSS 스타일 <필수> : [Object]

```
31 QCell1.setCellStyles(3, 5, 10, 7, {"background-color" : "red", "border-color" : "red"});
```

 See result

m-0007.html 실행 후 개발자 도구에서 아래 코드를 수행하여 확인

```
> qcell.setCellStyles(1, 2, 2, 5, {"background-color" : "#dd9955", "border-color" : "red"});
< true
> |
```

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

8) removeCellStyle

지정한 셀에 설정된 "셀 단위 스타일"을 제거하는 API 입니다.

 Syntax & eXample

```
removeCellStyle(row, col) → {boolean}
```

row: 로우 인덱스 <필수> : [number]

col: 컬럼 인덱스 <필수> : [number]

```
26 QCell1.removeCellStyle(3, 5);
```

 See result

setCellStyles 실행 후 개발자 도구에서 아래 코드를 수행하여 확인

```
> qcell.removeCellStyle(1,3)
< true
> qcell.removeCellStyle(2,4)
< true
> |
```

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

9) removeCellStyles

QCELL 은 지정한 범위 내의 셀들에 설정된 "셀 단위 스타일"을 제거하는 API 를 제공합니다.

 Syntax & eXample

```
removeCellStyles(firstrow, firstcol, lastrow, lastcol) → {boolean}
```

firstrow: 제거할 범위의 시작 로우 인덱스 <필수> : [number]

firstcol: 제거할 범위의 시작 컬럼 인덱스 <필수> : [number]

lastrow: 제거할 범위의 마지막 로우 인덱스 <필수> : [number]

lastcol: 제거할 범위의 마지막 컬럼 인덱스 <필수> : [number]

```
26 QCell1.removeCellStyles(3, 5, 10, 7);
```

 See result

setCellStyles 실행 후 removeCellStyles 에 원하는 입력 값을 넣어 실습해 보기 바랍니다.

10) clearCellStyles

적용된 모든 "셀 단위 스타일"을 제거하는 API 입니다.

 Syntax & eXample

```
clearCellStyles() → {boolean}
```

```
26 QCell1.clearCellStyles();
```

 See result

setCellStyles 실행 후 removeCellStyles 에 원하는 입력 값을 넣어 실습해 보기 바랍니다.

11) clearDynamicStyles

QCELL 객체 생성 이후 적용된 모든 스타일 변경사항을 초기화하는 API 입니다. ("컬럼 단위 스타일", "셀 단위 스타일", "highlightrepeat")을 초기화 합니다.

 Syntax & eXample

```
clearDynamicStyles() → {boolean}
```

```
26 QCell1.clearDynamicStyles();
```

 See result

동적으로 스타일을 적용 후 clearDynamicStyles 를 실행해 보기 바랍니다.

8. Column

8.1. 넓이 설정

컬럼에 대한 넓이 설정은 아래와 같은 규칙이 적용됩니다.

- 1) QCELL 는 각 컬럼의 넓이 값을 설정할 수 있습니다.
- 2) 컬럼의 넓이 설정 시 "px"과 "%"를 혼용해서 사용할 수 없습니다.
- 3) 넓이 값을 %로 설정하였을 때 전체 합이 100%가 넘어가면 자동으로 100%에 맞춰 재계산되어 화면에 출력됩니다.
- 4) 고정치의 단위는 px 이며, 고정 치로 값을 설정하였을 때 전체 합이 QCELL 의 넓이 값보다 클 경우 하단에 좌우 스크롤바가 생성됩니다
- 5) 넓이에 대한 설정을 하지 않았을 경우 기본 값 100px로 설정됩니다

Syntax & eXample

```

27     columns : [
28         {key: 'id', width: '200', title: ['카테고리ID'], type: "input", styleclassname: {"data"
29           , {key: 'date', width: '200', title: ['등록날짜'], type: "datepicker", styleclassname:
30           , {key: "name", width:'300',title: ['카테고리명'], type: "input", sort: true, stylecl
31           , {key: 'useYn', width: '150', title: ['사용여부'], type: "input", type:'selectmenu', s
32           , {key:"description", width:'300', title:['설명'], tooltip: true, type: 'textarea', re
33           , {key:"regUser", width: '150', title: ['등록자'], type: "input", sort: true, stylec
34     ],
35     width: "100%",
36     height: "245px"

```

See result

m-0008.html 을 실행, width 를 100%로 하는 경우와 특정 pixel 로 정의하여 QCELL 객체를 생성하는 경우의 차이를 확인하는 것으로 아래는 브라우저 넓이만큼 QCELL 객체의 넓이를 100%로 잡은 경우의 모습입니다.

<input type="checkbox"/>	카테고리ID	등록날짜	카테고리명	사용여부
<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y
<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	N
<input type="checkbox"/>	FOOD-00003	2019-04-10	우유류	Y
<input type="checkbox"/>	FOOD-00004	2019-04-10	통조림류	Y

해당 아래는 QCELL 객체의 크기를 810px 로 고정한 것으로 브라우저의 넓이에 무관하게 생성되는 모습을 확인할 수 있습니다.

<input type="checkbox"/>	카테고리ID	등록날짜	카테고리명	
<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	
<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	
<input type="checkbox"/>	FOOD-00003	2019-04-10	우유류	
<input type="checkbox"/>	FOOD-00004	2019-04-10	통조림류	

check eXample 전체 소스는 manual 내 "QCELL 객체의 넓이, 각 컬럼의 넓이 설정 동작에 대한 확인" 참조

More info. API 문서의 create 를 참조

8.2. 지원 유형

QCELL 은 아래와 같이 다양한 column 유형을 지원합니다.

- static: 조회를 위한 수정 불가의 기본 유형(default)
- input: 입력, 수정, 삭제가 가능한 유형
- selectmenu: 콤보 박스
- checkbox: 체크박스과 라디오 형태를 같이 사용할 수 있는 유형
- datepicker: jquery-ui 의 datepicker 를 사용할 수 있는 유형
- image: 이미지를 사용할 수 있는 유형
- textarea: 입력, 수정, 삭제가 가능한 textarea 유형
- html(renderer 타입): 출력만이 가능한 사용자 정의 함수 유형(수정 불가)
-

Syntax & eXample

```

27 columns : [
28   {key: 'name', type: "input", width: '10%', title: ['기본정보', '이름'], styleclassname: {"data-
29     , {key: "id", title: ['기본정보', "고객번호"], sort: true, width: '8%', styleclassname: {"data-
30     , {key: 'cost', type: "input", width: '8%', title: ['상세정보', '년간 회비'], styleclassname:
31     , {key: "contact_date", type: 'datepicker', title: ['상세정보', '연락일'], sort: true, width: '8
32     , {key: "contact_method", type: 'selectmenu', title: ['연락정보', '연락방법'], sort: true, wid
33     , {key: 'phone', width: '10%', title: ['연락정보', '전화번호'], styleclassname: {"data": "al
34     , {key: 'email', type: "html", width: '12%', title: ['연락정보', '이메일'], options: {html:
35     , {key: 'address', type: "textarea", width: '30%', title: ['연락정보', '주소'], tooltip: true
36     , {key: "country", type: "image", title: ['연락정보', "국가"], disabled: false, options: {
37     ]

```

See result

m-0009.html 의 다양한 컬럼 유형으로 QCELL 객체를 생성하여 확인

	기본정보		상세정보		연락정보				
	이름	고객번호	연간 회비	연락일	연락방법	전화번호	이메일	주소	국가
1	김철수	21-51...	₩ 1,2...	2018...	전화	82-2-345...	✉ bancboy@ac	한국 서울 종로구 태평로 123	
2	けん たか...	21-52...	₩ 1,1...	2018...	이메일	81-3)345...	✉ jmgomez@f	東京都港区南麻布1-2-5	
3	আব্দুল হা...	11-51...	₩ 1,2...	2018...	전화	(+880) 1...	✉ gamma@me	4 মাদানী এভিনিউ, বারিধারা, ঢাকা -1212	
4	Nguyễn P...	21-71...	₩ 900...	2018...	이메일	+84-24) ...	✉ yamla@mac	28th Fl., Lotte Center Hà Nội, 54 Liễu...	
5	上班族	11-51...	₩ 1,3...	2018...	우편	(86-10) 8...	✉ drotsky@opt	北京市人民政府办公厅	

check eXample 전체 소스는 manual 내 "다양한 컬럼 유형 설정과 동작에 대한 확인" 참조

More info. API 문서의 create 영역의 columns 내 type 를 참조

8.3. 컬럼 고정

QCELL 객체는 컬럼 고정을 지원합니다. 컬럼의 수가 많아서 한 화면에 모든 컬럼이 표현되지 않을 때 좌우 스크롤 바가 생성되고 우측으로 이동시 좌측의 컬럼은 화면에서 사라지게 되는데 이 때 함께 봐야 할 중요한 컬럼인 경우 고정시켜 계속 보이게 할 때 사용합니다.

Syntax & eXample

```
29     frozencols : 1,
30     frozenrows : 0,
31     frozenbottomrows : 0,
```

See result

m-0008.html 을 실행하여 좌우 스크롤바를 이동해 보면서 확인

<input type="checkbox"/>	카테고리ID	사용여부	설명	등록자
<input type="checkbox"/>	FOOD-00001	Y	라면제품에 대한 카테고리입니다.	홍길동
<input type="checkbox"/>	FOOD-00002	N	주류에 대한 카테고리입니다.	김길동
<input type="checkbox"/>	FOOD-00003	Y	우유류와 유제품에 대한 카테고리입니다.	박길동
<input type="checkbox"/>	FOOD-00004	Y	캔으로된 통조림에 대한 카테고리입니다.	최길동

8.4. 체크 박스

체크 박스를 지원하는 것은 크게 로우 헤더에 넣는 것과 컬럼의 타입으로 넣은 것이

있으며 컬럼의 타입으로 넣을 data 바인딩을 통해 값을 설정하고 그 값을 서버에 전달하는 등의 로직 개발이 편리합니다. 여기서는 컬럼 타입으로 체크 박스를 지원하는 것에 대하여 설명합니다.

Data 영역의 체크 박스는 각 로우에 대한 체크 여부의 값을 갖는 것이고 header 영역의 체크 박스는 모든 로우에 대한 전체 선택, 전체 해제 기능을 수행합니다.

- header 영역은 별개의 영역이기 때문에 header 에 checkbox 를 사용하기 위해서는 column 선언 시 속성을 추가하여 상단 체크박스를 사용하겠다고 선언해야 합니다.

1) wholeselect

header 영역에 전체 체크박스 사용 여부를 설정합니다.

Syntax & eXample

컬럼 정의 내 options: {wholeselect: **true** 여부}

정의하지 않으면 기본적으로 체크 박스 컬럼에 대한 헤더 체크 박스가 생성되지 않습니다.

```
28 {title: ["T/F체크"], key: "check1", type: "checkbox", options: {wholeselect: true, checkedvalue: "T", uncheckedvalue: "F"}},
29 {title: ["OK/ng"], key: "check2", type: "checkbox", options: {checkedvalue: "ok", uncheckedvalue: "ng"}},
30 {title: ["1/0"], key: "check3", type: "checkbox", options: {checkedvalue: "1", uncheckedvalue: "0"}},
31 {title: ["라디오 체크"], key: "check4", type: "checkbox", options: {likeradio: true}},
```

See result

m-0010.html 을 실행하여 상단의 체크 박스 유무 확인

	<input type="checkbox"/>	<input checked="" type="checkbox"/>	OK/ng	1/0	라디오 체크	firstname	lastname	age	phone
1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Bobbie	Goodman	44	870-4412
2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="radio"/>	Mclaughlin	Conrad	49	938-4902

2) setHeaderChecked

QCELL 은 체크박스 컬럼의 헤더에 속성이 전체 체크를 제공하고 있을 경우 API 를 통해 전체 선택과 해제를 동작 시킬 수 있습니다.

Syntax & eXample

```
setHeaderChecked(col, state) → {void}
```

col: 컬럼 인덱스

state: 전체 선택 체크박스 상태 값 "true" | "false"

```
27 QCell1.setHeaderChecked(1, true);
```

See result

m-0010.html 를 실행 후 개발자 도구에서 아래 코드를 통해 API 로 체크 박스에 대한 전체 선택과 해제를 확인

```
> qcell.setHeaderChecked(2, true)
< undefined
> qcell.setHeaderChecked(2, false)
< undefined
> |
```

3) getHeaderCheckedState

전체 선택 체크박스가 존재할 경우, 해당 컬럼의 전체 선택 체크박스의 상태 값을 반환하는 API 입니다.

 **Syntax & eXample**

```
getHeaderCheckedState(col) → {boolean}
```

col: 컬럼 인덱스

```
27 QCell11.getHeaderCheckedState(1);
```

 **See result**

m-0010.html 를 실행 후 개발자 도구에서 아래 코드를 통해 API 로 체크 박스에 대한 전체 선택과 해제 시 값을 확인할 수 있습니다.

```
> qcell.getHeaderCheckedState(2)
< false
> qcell.setHeaderChecked(2, true)
< undefined
> qcell.getHeaderCheckedState(2)
< true
> |
```

8.5. 컬럼 추가

QCELL 은 컬럼 추가를 지원합니다.

1) insertColumn

컬럼을 추가합니다.

 **Syntax & eXample**

```
insertColumn(nCol, objCol) → {boolean}
```

nCol : 추가할 컬럼 위치 <필수> : [number]

objCol : 추가할 컬럼 정보 <필수> : [object]



manual 폴더 내의 m-0028.html 샘플 파일을 실행해주세요.

이름	전화번호	이메일	사명	입사일	주소
고길동	010-1234-5678	gogildong@gamil.comm	길동전자	1978-02-03	서울 특별시 송파구
둘리	010-2345-6789	twori@gamil.comm	길동전자	1992-07-21	서울 특별시 송파구
마이콜	010-3456-7890	lamen@gamil.comm	신흥음악학원	1991-02-15	서울 특별시 송파구
도우너	010-1111-2222	dunkin@gamil.comm	길동전자	1992-07-21	서울 특별시 송파구
또치	010-2222-3456	again@gamil.comm	길동전자	1992-07-21	서울 특별시 송파구
박희동	010-1234-5678	hidong@gamil.comm	길동전자	입사 예정	서울 특별시 송파구

이름	나이	전화번호	이메일	사명	입사일
고길동		010-1234-5678	gogildong@gamil.comm	길동전자	1978-02-03
둘리		010-2345-6789	twori@gamil.comm	길동전자	1992-07-21
마이콜		010-3456-7890	lamen@gamil.comm	신흥음악학원	1991-02-15
도우너		010-1111-2222	dunkin@gamil.comm	길동전자	1992-07-21
또치		010-2222-3456	again@gamil.comm	길동전자	1992-07-21
박희동		010-1234-5678	hidong@gamil.comm	길동전자	입사 예정

nCol 은 0, 1, 2, 3, 4... 순으로 설정하여 동작합니다.

check eXample 전체 소스는 [View Source](#) 버튼을 클릭하여 참조

8.6. 컬럼 삭제

QCELL 은 컬럼 삭제를 지원합니다.

1) removeColumn

컬럼을 삭제합니다.

JS Syntax & eXample

`removeColumn(nCol) → {boolean}`

nCol : 삭제할 컬럼 정보 <필수> : [number]

See result

manual 폴더 내의 m-0028.html 샘플 파일을 실행해주세요.

이름	전화번호	이메일	사명	입사일	주소
고길동	010-1234-5678	gogildong@gamil.comm	길동전자	1978-02-03	서울 특별시 송파구
둘리	010-2345-6789	twori@gamil.comm	길동전자	1992-07-21	서울 특별시 송파구
마이클	010-3456-7890	lamen@gamil.comm	실용음악학원	1991-02-15	서울 특별시 송파구
도우너	010-1111-2222	dunkin@gamil.comm	길동전자	1992-07-21	서울 특별시 송파구
또치	010-2222-3456	again@gamil.comm	길동전자	1992-07-21	서울 특별시 송파구
박희동	010-1234-5678	hidong@gamil.comm	길동전자	입사 예정	서울 특별시 송파구

컬럼 추가 | 컬럼 삭제 | html로 저장

전화번호	이메일	사명	입사일	주소
010-1234-5678	gogildong@gamil.comm	길동전자	1978-02-03	서울 특별시 송파구
010-2345-6789	twori@gamil.comm	길동전자	1992-07-21	서울 특별시 송파구
010-3456-7890	lamen@gamil.comm	실용음악학원	1991-02-15	서울 특별시 송파구
010-1111-2222	dunkin@gamil.comm	길동전자	1992-07-21	서울 특별시 송파구
010-2222-3456	again@gamil.comm	길동전자	1992-07-21	서울 특별시 송파구
010-1234-5678	hidong@gamil.comm	길동전자	입사 예정	서울 특별시 송파구

컬럼 추가 | 컬럼 삭제 | html로 저장

nCol 은 0, 1, 2, 3, 4... 순으로 설정하여 동작합니다.

✓check eXample 전체 소스는 View Source 버튼을 클릭하여 참조

8.7. 컬럼 초기화

QCELL 은 컬럼 초기화를 지원합니다.

1) resetColumns

컬럼을 초기화합니다.

 Syntax & eXample

```
resetColumns() → {boolean}
```

 See result

manual 폴더 내의 m-0028.html 샘플 파일을 실행해주세요.

주소
서울 특별시 송파구

이름	전화번호	이메일	사명	입사일	주소
고길동	010-1234-5678	gogildong@gamil.comm	길동전자	1978-02-03	서울특별시 송파구
둘리	010-2345-6789	twori@gamil.comm	길동전자	1992-07-21	서울특별시 송파구
마이콜	010-3456-7890	lamen@gamil.comm	실용음악학원	1991-02-15	서울특별시 송파구
도우너	010-1111-2222	dunkin@gamil.comm	길동전자	1992-07-21	서울특별시 송파구
또치	010-2222-3456	again@gamil.comm	길동전자	1992-07-21	서울특별시 송파구
박희동	010-1234-5678	hidong@gamil.comm	길동전자	입사 예정	서울특별시 송파구

컬럼 추가 컬럼 삭제 **컬럼 초기화** html로 저장

컬럼이 추가 및 삭제가 된 그리드의 컬럼을 초기화합니다.

 **check eXample** 전체 소스는 [View Source](#) 버튼을 클릭하여 참조

8.8. 컬럼 합계

QCELL 은 합계를 지원합니다.

1) getColSum

특정 컬럼의 데이터를 모두 더합니다.

 **Syntax & eXample**

```
getColSum(nCol) → {Number}
```

nCol : 컬럼의 열 인덱스 <필수> : [number]

```
qcell.getColSum(4)
```

 **See result**

manual 의 m-0016.html 파일을 실행하여 개발자 도구를 열어 아래 코드를 입력해주세요.

```
> qcell.getColSum(4)
< 4052
```

기본정보				상세정보						
지역	구분	유형	수량	면적	비용	주소	면적	시		
강원도	사유림	필성자연휴양림	300	498,263	속박시설 이용객만 입장 가...	강원도 필성군 갑천면 정포로 430번길 113	033-343-3392	속속리길, 산림문		
강원도	사유림	주천강자연휴양림	400	2,296,100	일반 : 2000원, 청소년 : 1000원	강원도 필성군 강변로 영랑6길 115	033-345-7575	속속리길, 산림문		
강원도	국유림	솔채원	300	153,710	별도 입장으로 없음	강원도 필성군 동내면 청태산로 777	033-340-6300	탐방로, 속박동,		
강원도	공유림	치악산자연휴양림	700	1,700,000	일반(개인 2000, 청소년 및 ...	강원도 원주시 판부면 휴양림길 66	033-762-9288	속속리길, 주차장		
강원도	공유림	동강간방자연휴양림	200	448,966	없음	강원도 정선군 신동읍 동강로 916-212	033-560-3464	동강간방자연휴양		
강원도	국유림	태백고원자연휴양림	126	1,100,000	일반2000원+단체 1500원	강원도 태백시 머리골길 153	033-582-7440	산림문화휴양관		
가희도	사유림	희서지여수안방	300	498,263	소박시설 이용객만 입장 가...	가희도 희서구 가희면 거포로 339번길 113	033-343-3392	속속리길, 산림문		
			평균: 338명	총면적: 12,406,078	해당사항없음	해당사항없음	해당사항없음	해당사항없음		

4 열에 있는 수용인원 컬럼의 합계한 결과값인 4052 를 확인할 수 있습니다.

8.9. 셀렉트 메뉴 컬럼과 label

1) setSelectedMenuValue

selectmenu Column 의 데이터에서 value 값을 설정하여, 그에 해당하는 label 값을 QCELL 에 출력시킵니다.

JS Syntax & eXample

```
setSelectedMenuValue(row, col, val) → {boolean}
```

row : 행 인덱스 <필수> : [number]

col : 열 인덱스 <필수> : [number]

val : 설정값 <필수> : [string | number]

```
qcell.setSelectedMenuValue(1, 4, 'Y')
```

See result

manual 의 m-0005.html 파일을 실행하여 개발자 도구를 열어 아래 코드를 입력해주세요.

```
> qcell.setSelectedMenuValue(1,4, 'Y')
< true
```

카테고리ID	등록일자	카테고리명	사용여부	설명	등록자
FOOD-00001	2019-04-10	라면류	미사용	라면제품에 대한 카테고리입니다.	홍길동
FOOD-00002	2019-04-10	주류	미사용	주류에 대한 카테고리입니다.	김길동
FOOD-00003	2019-04-10	우유류	사용	우유류와 유제품에 대한 카테고리입니다.	박길동
FOOD-00004	2019-04-10	통조림류	사용	캔으로된 통조림에 대한 카테고리입니다.	최길동



카테고리ID	등록일자	카테고리명	사용여부	설명	등록자
FOOD-00001	2019-04-10	라면류	사용	라면제품에 대한 카테고리입니다.	홍길동
FOOD-00002	2019-04-10	주류	미사용	주류에 대한 카테고리입니다.	김길동
FOOD-00003	2019-04-10	우유류	사용	우유류와 유제품에 대한 카테고리입니다.	박길동
FOOD-00004	2019-04-10	통조림류	사용	캔으로된 통조림에 대한 카테고리입니다.	최길동

value 에 저장된 'Y' 값을 선택하여 이에 해당하는 label 값인 '사용'을 그리드에 표시해줍니다.

2) getSelectmenuLabel

selectmenu Column 의 value 값으로 label 값을 가져오는 API 입니다.

Syntax & eXample

```
getSelectmenuLabel(row, col) → {string}
```

row : 행 인덱스 <필수> : [number]

col : 열 인덱스 <필수> : [number]

8.10. 필수 컬럼 정보 반환

1) checkRequired

required 컬럼을 확인하여 비어 있다면 정보를 반환합니다.

Syntax & eXample

```
checkRequired() → {Array}
```

```
qcell.checkRequired()
```

See result

manual 의 m-0027.html 파일을 실행해주세요.

*해당 옵션을 사용하기 위해선 컬럼 options 으로 required : true 을 설정하여야합니다.

```
options: { required: true }
```

<결과값>

순번	등록번호	기업			
		기업규모	업체명	소재지	배정인원
1	1	과학진흥연구소	에피지놈제어연구센터(성균관대)	대한민국 경기도 어딘가의 수원시에서도 가장 큰 장안구 서부로 2066에서도 더 들어가서 우회선하고 좌회선 후 직진	5
2	7	국가기관 등 연구소			7
3	5	국가기관 등 연구소	나노튜브및나노복합구조연구센터(성균관대)	경기도 수원시 장안구 서부로 2066 Ncenter 86371호	7

required 필드 확인 add a row

순번	등록번호	기업			
		기업규모	업체명	소재지	배정인원
1	1	과학진흥연구소	에피지놈제어연구센터(성균관대)	대한민국 경기도 어딘가의 수원시에서도 가장 큰 장안구 서부로 2066에서도 더 들어가서 우회선하고 좌회선 후 직진	5
2	7	국가기관 등 연구소			7
3	5	국가기관 등 연구소	나노튜브및나노복합구조연구센터(성균관대)	경기도 수원시 장안구 서부로 2066 Ncenter 86371호	7

값이 보여질 영역

3행의 3열 / 3행의 4열 /

required 필드 확인 버튼을 누르면 값이 비어있는 컬럼의 정보를 반환합니다.

```
> qcell.checkRequired()
< (2) [{"row": 3, "col": 3}, {"row": 3, "col": 4}]
  length: 2
  [[Prototype]]: Array(0)
```

개발자 도구의 콘솔창을 열어 해당 API 를 입력하여 확인해주세요.

8.11. 특정 위치 이동

1) moveCol

QCELL 은 입력받은 열 인덱스에 해당하는 컬럼을 특정 위치로 이동시키는 API 입니다.

Syntax & eXample

```
moveCol(beforecol, aftercol) → {boolean}
```

beforecol : 이동하기 전의 열 인덱스 <필수> : [number]

aftercol: 이동한 후의 열 인덱스 <필수> : [number]

```
qcell.moveCol(1, 3);
```

 See result

manual 의 m-0026.html 파일 실행해주세요.

*해당 옵션을 사용하기 위해선 컬럼의 이동이 가능하도록 `move : true` 을 설정하여야합니다.

<결과값>

```
columns:
  [
    { "width": "20%", "key": "name", "title": ["이름"], move:true },
    { "width": "10%", "key": "phone", "title": ["전화번호"], move:true },
    { "width": "20%", "key": "email", "title": ["이메일"], move:true },
    { "width": "20%", "key": "company", "title": ["사명"], move:true },
    { "width": "10%", "key": "regdate", "title": ["입사일"], options: { format: {origin: "YYYY-MM-DD"}}, move:true },
    { "width": "20%", "key": "addr", "title": ["주소"], move:true }
  ]
};
```

이름	전화번호	이메일	사명	입사일	주소
홍길동	010-1111-2222	qwer@gmail.com	길동상가	20190813	서울특별시 송파구 잠실동
임격정	010-2222-3333	qwer@gmail.com	길동상가	20190813	서울특별시 송파구 잠실동
심청이	010-4444-5555	qwer@gmail.com	길동상가	20190813	서울특별시 송파구 잠실동
문익점	010-6666-7777	qwer@gmail.com	길동상가	20190813	서울특별시 송파구 잠실동
황건적	010-8888-9999	qwer@gmail.com	길동상가	20190813	서울특별시 송파구 잠실동

<결과값>

이름	이메일	사명	전화번호	입사일	주소
홍길동	qwer@gmail.com	길동상가	010-1111-2222	20190813	서울특별시 송파구 잠실동
임격정	qwer@gmail.com	길동상가	010-2222-3333	20190813	서울특별시 송파구 잠실동
심청이	qwer@gmail.com	길동상가	010-4444-5555	20190813	서울특별시 송파구 잠실동
문익점	qwer@gmail.com	길동상가	010-6666-7777	20190813	서울특별시 송파구 잠실동
황건적	qwer@gmail.com	길동상가	010-8888-9999	20190813	서울특별시 송파구 잠실동

8.12. 컬럼 정보

1) getColumnInfo

현재 컬럼의 정보를 얻습니다.



```
getColumnInfo(nCol) → {object}
```

nCol : 정보를 얻을 컬럼 정보 <필수> : [number]

```
qcell.getColumnInfo(1)
```

 See result

<결과값>

```
> qcell.getColumnInfo(1)
<
  ▾ {width: '10%', key: 'phone', title: Array(1), move: true}
    key: "phone"
    move: true
    ▾ title: Array(1)
      0: "전화번호"
      length: 1
      ▶ [[Prototype]]: Array(0)
    width: "10%"
    ▶ [[Prototype]]: Object
```

2) getCurrentColumnsInfo

현재 컬럼의 최신 정보를 얻습니다.

 Syntax & eXample

`getCurrentColumnsInfo()` → {Array}

<결과값>

```
> qcell.getCurrentColumnsInfo()
<
  ▾ (6) [{"...", "...", "...", "...", "...", "..."} ⓘ
    ▾ 0:
      data_text_align: "left"
      datatype: "string"
      disabled: false
      editor: false
      key: "name"
      move: true
      ▶ options: {}
      remove: false
      resize: false
      search: false
      sort: false
      ▶ style: {}
      ▶ styleclassname: {header: '', data: ''}
      ▶ title: ['이름']
      tooltip: false
      type: "static"
      width: "304px"
      ▶ [[Prototype]]: Object
      ▶ 1: {title: Array(1), type: 'static', width: '152px', key: 'phone', styleclassname: {...}, ...}
      ▶ 2: {title: Array(1), type: 'static', width: '304px', key: 'email', styleclassname: {...}, ...}
      ▶ 3: {title: Array(1), type: 'static', width: '304px', key: 'company', styleclassname: {...}, ...}
      ▶ 4: {title: Array(1), type: 'static', width: '152px', key: 'regdate', styleclassname: {...}, ...}
      ▶ 5: {title: Array(1), type: 'static', width: '302px', key: 'addr', styleclassname: {...}, ...}
    length: 6
```

9. Row

9.1. 틀 고정

컬럼 고정은 컬럼이 많은 로우 데이터에서 특정 컬럼의 데이터를 항상 보면서 사용하기 위한 편리함이라면 로우 고정은 많은 로우 데이터 중에서 상단에 있거나 하단에 있는 데이터를 고정해 놓고 사용하기 위한 편리함을 제공합니다.

1) frozenrows

지정한 수만큼 상단의 로우가 고정되어 상하 스크롤이 되도 항상 상단에 노출됩니다.

Syntax & eXample

"frozenrows" : number,

```
29     frozencols : 1,
30     frozenrows : 1,
31     frozenbottomrows : 1,
```

See result

m-0011.html 을 실행하여 상하 스크롤링하면 1 번째 로우는 화면에서 계속 보여지는 것을 확인

	<input type="checkbox"/>	카테고리ID	등록날짜	카테고리명	
1	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	
4	<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	
5	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	
6	<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	
7	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	
8	<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	
9	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	
14	<input type="checkbox"/>	FOOD-00004	2019-04-10	통조림류	

 **More info.** API 문서의 create 영역의 property 참조

2) frozenbottomrows

지정한 수만큼 마지막 로우부터 하단 영역 ss 고정됩니다.

Syntax & eXample

"frozenbottomrows" : number,

```
29     frozencols : 1,
30     frozenrows : 1,
31     frozenbottomrows : 1,
```

 See result

m-0011.html 을 실행하여 상하 스크롤링하면 14 번째 로우는 화면에서 계속 보여지는 것을 확인

	<input type="checkbox"/>	카테고리ID	등록날짜	카테고리명
1	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류
4	<input type="checkbox"/>	FOOD-00002	2019-04-10	주류
5	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류
6	<input type="checkbox"/>	FOOD-00002	2019-04-10	주류
7	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류
8	<input type="checkbox"/>	FOOD-00002	2019-04-10	주류
9	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류
14	<input type="checkbox"/>	FOOD-00004	2019-04-10	통조림류

 More info. API 문서의 create 영역의 property 참조

3) setFrozenRows

QCELL 은 고정행을 새로 설정하는 API 를 제공합니다.

 Syntax & eXample

```
setFrozenRows(nIndex) → {boolean}
```

nIndex: 타입을 반환할 열 인덱스 <필수> : [number]

```
qcell.setFrozenRows(4)
```

 See result

m-0011.html 을 실행하여 개발자 도구를 열어 아래 명령어를 사용해보세요

```
qcell.setFrozenRows(4)
true
```

카테고리ID	등록일자	카테고리명	사용여부	설명
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.

기존 1 행의 고정행에서 4 행까지의 고정행으로 새로 설정하는 것을 확인

카테고리ID	등록일자	카테고리명	사용여부	설명	등록자
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍길동
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍길동
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍길동
FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.	김길동
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍길동
FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.	김길동

4) getFrozenRows

QCELL 은 고정행 인덱스를 반환하는 API 를 제공합니다.

JS Syntax & eXample

```
getFrozenCols() → {Number}
```

```
qcell.getFrozenRows();
```

See result

m-0011.html 을 실행하여 개발자 도구를 열어 qcell.setFrozenRows(4) 명령어를 입력하여

카테고리ID	등록일자	카테고리명	사용여부	설명
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.

기존 1 행의 고정행에서 4 행까지의 고정행으로 새로 설정하는 것을 확인 후

카테고리ID	등록일자	카테고리명	사용여부	설명	등록자
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍길동
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍길동
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍길동
FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.	김길동
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍길동
FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.	김길동

아래 명령어를 입력하여 확인

```
qcell.getFrozenRows();
```

```
4
```

5) setFrozenCols

QCELL 은 고정열을 새로 설정하는 API 를 제공합니다.

Syntax & eXample

```
setFrozenCols(nIndex) → {boolean}
```

nIndex: 타입을 반환할 열 인덱스 <필수> : [number]

```
qcell.setFrozenCols(3);
```

See result

m-0011.html 을 실행하여 개발자 도구를 열어 아래 코드를 입력해주세요

```
qcell.setFrozenCols(3);  
true
```

카테고리ID	등록일자	카테고리명	사용여부	설명
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.
FOOD-00004	2019-04-10	통조림류	Y	캔으로된 통조림에 대한 카테고리입니다.

기존 1 열의 고정열에서 3 열까지의 고정열로 새로 설정하는 것을 확인

카테고리ID	등록일자	카테고리명	사용여부	설명
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00004	2019-04-10	통조림류	Y	캔으로된 통조림에 대한 카테고리입니다.

6) getFrozenCols

QCELL 은 고정열 인덱스를 반환하는 API 입니다.

Syntax & eXample

```
getFrozenCols() → {Number}
```

See result

m-0011.html 을 실행하여 개발자 도구를 열어 qcell.setFrozenCols(); 명령어를 입력해주세요.

카테고리ID	등록일자	카테고리명	사용여부	설명
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.
FOOD-00004	2019-04-10	통조림류	Y	캔으로된 통조림에 대한 카테고리입니다.

기존 1 열의 고정열에서 3 열까지의 고정열로 새로 설정하는 것을 확인 후

카테고리ID	등록일자	카테고리명	사용여부	설명
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.
FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
FOOD-00004	2019-04-10	통조림류	Y	캔으로된 통조림에 대한 카

아래 코드를 입력하여 확인해주세요

```
qcell.getFrozenCols();
3
```

7) setFrozenBottomRows

QCELL 은 하단 고정행 새로 설정하는 API 입니다.

Syntax & eXample

```
setFrozenBottomRows(nIndex) → {boolean}
```

nIndex: 타입을 반환할 열 인덱스 <필수> : [number]

```
qcell.setFrozenBottomRows(3);
```

See result

m-0011.html 을 실행하여 개발자 도구를 열어 아래 명령어를 입력해주세요.

```
qcell.setFrozenBottomRows(3);
true
```

	<input type="checkbox"/>	카테고리ID	등록날짜	카테고리명	사용여부	설명	등록자
1	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍길동
2	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍길동
3	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍길동
4	<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.	김길동
5	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍길동
6	<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.	김길동
7	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍길동
14	<input type="checkbox"/>	FOOD-00004	2019-04-10	통조림류	Y	캔으로된 통조림에 대한 카테고리입니다.	최길동

기존 틀고정의 하단 1 행 기준에서 3 행 기준으로 바뀌는 것을 확인

	<input type="checkbox"/>	카테고리ID	등록날짜	카테고리명	사용여부	설명	등록자
1	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍길동
5	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍길동
6	<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.	김길동
7	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍길동
8	<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.	김길동
12	<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.	김길동
13	<input type="checkbox"/>	FOOD-00003	2019-04-10	우유류	Y	우유류와 유제품에 대한 카테고리입니다.	박길동
14	<input type="checkbox"/>	FOOD-00004	2019-04-10	통조림류	Y	캔으로된 통조림에 대한 카테고리입니다.	최길동

8) getFrozenBottomRows

QCELL 은 하단 고정행 인덱스를 반환하는 API 입니다.

Syntax & eXample

```
getFrozenBottomRows() → {Number}
```

```
qcell.getFrozenBottomRows();
```

See result

m-0011.html 을 실행하여 개발자 도구에 qcell.setFrozenBottomRows() 코드를 입력해주세요

아래 코드를 입력하여 확인해주세요

```
qcell.getFrozenBottomRows();  
3
```

9.2. 행 구분

로우가 많은 경우 유관으로 데이터를 식별하기 쉽도록 반복되는 로우에 대한 구분과 함께 스타일을 다르게 지정할 수 있습니다.

1) highlightrepeat

로우 스타일을 반복하여 적용하는 "highlightrepeat" 속성을 설정합니다.

 Syntax & eXample

"highlightrepeat" : number,

```
29 highlightrepeat : 0,
```

 See result

m-0011.html 의 highlightrepeat 를 0 으로 설정 후 실행하여 모습 확인해주세요.

	<input type="checkbox"/>	카테고리ID	등록날짜	카테고리명	
1	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	
2	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	
3	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	
4	<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	
5	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	
6	<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	
7	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	
14	<input type="checkbox"/>	FOOD-00004	2019-04-10	통조림류	

highlightrepeat 가 3 일 때 모습 확인할 수 있습니다.

	<input type="checkbox"/>	카테고리ID	등록날짜	카테고리명	
1	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	
2	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	
3	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	
4	<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	
5	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	
6	<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	
7	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	
14	<input type="checkbox"/>	FOOD-00004	2019-04-10	통조림류	

2) getHighlightRepeat

QCELL 객체에 적용된 "highlightrepeat" 속성 값을 반환하는 API 입니다.

 Syntax & eXample

```
getHighlightRepeat() → {number}
```

```
29 var nHighlightRepeat = QCell1.getHighlightRepeat();
```

 See result

m-0011.html 의 highlightrepeat 를 0 으로 설정 후 개발자 도구에서 실행

```
> qcell.getHighlightRepeat()
< 0
```

3) setHighlightRepeat

QCELL 은 행 스타일을 반복하여 적용하는 "highlightrepeat" 속성의 주기를 설정하는 API 를 제공합니다.

Syntax & eXample

```
setHighlightRepeat(cycle) → {boolean}
```

cycle: 스타일 반복 주기 <필수> : [number]

```
31 QCell1.setHighlightRepeat(3);
```

See result

m-0011.html 의 highlightrepeat 를 0 으로 설정 후 개발자 도구에서 아래 내용을 실행하여 3 개의 로우 단위로 스타일이 적용되는지 확인

```
> qcell.setHighlightRepeat(3)
< true
>
```

 **More info.** QCELL 은 기본적으로 틀 고정과 highlightrepeat 에 대한 스타일이 정의되어 있어 위와 같이 보여집니다. 이를 다르게 적용하는 것은 스타일 부분을 참조하시기 바랍니다.

10. Col

10.1. key 속성

각 컬럼에 바인딩 된 Key 와 관련한 API 에 대한 사용법에 대한 설명입니다.

1) getColOfKey

QCELL 은 입력받은 "key" 속성값이 설정된 컬럼의 인덱스를 반환하는 API 입니다. 만약 Key 가 둘 이상의 컬럼에 바인딩 된 경우 "getColsOfKey" API 를 사용하고 이때는 여러 컬럼의 인덱스 값을 배열로 반환합니다.

 Syntax & eXample

`getColOfKey(key) → {Number}`

key: "key" 속성값 <필수> : [string]

```
31 QCell11.getColOfKey("name");
```

 See result

m-0011.html 을 실행 후 개발자 도구에서 아래 내용을 확인

```
> qcell.getColOfKey('name');
< 4
> |
```

2) getKeyOfCol

특정 컬럼에 바인딩 된 Key 를 알고 싶을 때 사용하는 API 입니다.

 Syntax & eXample

`getKeyOfCol(col) → {string}`

col: 컬럼 인덱스 <필수> : [number]

```
31 QCell11.getKeyOfCol(3);
```

 See result

m-0011.html 을 실행 후 개발자 도구에서 아래 내용을 확인

```
> qcell.getKeyOfCol(4);
< "name"
> |
```

10.2. 컬럼 ID/Type/Index 반환

1) getColID

입력받은 컬럼 인덱스의 컬럼 ID 를 반환하는 API 입니다. Html5 div tag 기반으로 각 로우, 컬럼이 정의되어 있는데 해당 컬럼에 대한 접근과 활동을 위해 ID 를 제공하고 있습니다.

 Syntax & eXample

```
getColID(col) → {string}
```

col: 컬럼 인덱스 <필수> : [number]

```
31 QCell1.getColID(3);
```

 See result

m-0011.html 을 실행 후 개발자 도구에서 아래 내용을 확인

```
> qcell.getColID(4)
< "col_4"
> |
```

2) getColOfID

입력받은 컬럼 ID 에 해당하는 컬럼 인덱스를 반환하는 API 입니다.

 Syntax & eXample

```
getColOfID(id) → {number}
```

id: 컬럼 ID <필수> : [string]

```
28 QCell1.getColOfID("col_3");
```

 See result

m-0011.html 을 실행 후 개발자 도구에서 아래 내용을 확인

```
> qcell.getColOfID("col_4");
< 4
> |
```

3) getColType

입력한 컬럼 인덱스의 컬럼 타입을 반환하는 API 입니다.

 Syntax & eXample

```
getColType(col) → {String}
```

col: 타입을 반환할 컬럼 인덱스 <필수> : [number]

```
28 QCell1.getColType(3);
```

 See result

m-0011.html 을 실행 후 개발자 도구에서 아래 내용을 확인

```
> qcell.getColType(4)
< "input"
> qcell.getColType(3)
< "datepicker"
> |
```

4) getColAttribute

QCELL 은 입력한 열 인덱스에 해당하는 열의 속성을 반환하는 API 를 제공합니다.

Syntax & eXample

```
getColAttribute(col, attribute) → {String}
```

col: 타입을 반환할 열 인덱스 <필수> : [number]

attribute: 가져올 속성 명칭 <필수> : [string]

```
qcell.getColAttribute(3,"width");
```

See result

m-0011.html 을 실행 후 개발자 도구에서 아래 내용을 확인

```
> qcell.getColAttribute(3,"width");
< '200'
> qcell.getColAttribute(4,"width");
< '300'
>
```

10.3. 넓이 설정 및 반환

1) getColWidth

입력한 컬럼 인덱스에 해당하는 컬럼의 너비를 반환하는 API 입니다.

Syntax & eXample

```
getColWidth(col) → {number}
```

col: 너비를 반환할 컬럼 인덱스 <필수> : [number]

```
28 QCell1.getColWidth(3);
```

See result

m-0011.html 을 실행 후 개발자 도구에서 아래 내용을 확인

```
> qcell.getColWidth(3)
< 200
> qcell.getColWidth(4)
< 300
> |
```

2) setColWidth

입력한 컬럼 인덱스에 해당하는 컬럼의 넓이를 변경하는 API 입니다. 이때 아래와 같은 제약 조건이 있습니다.

* 15 보다 작은 값으로 넓이 값을 설정할 수 없습니다.

* 컬럼의 넓이는 px 단위로만 변경할 수 있습니다.

Syntax & eXample

`setColWidth(col, colwidth) → {boolean}`

col: 컬럼 인덱스 <필수> : [number]

colwidth : 컬럼의 넓이 설정 <필수> : [number] (최소값: 15)

```
28 QCell11.setColWidth(3, 200);
```

See result

m-0011.html 을 실행 후 개발자 도구에서 3 번째, 4 번째 컬럼을 같은 100px 로 설정한 후 내용을 확인

```
> qcell.setColWidth(3, 100)
< undefined
> qcell.setColWidth(4, 100)
< undefined
> |
```

	<input type="checkbox"/>	카테고리ID	등록날짜	카테고리명	사용여부	설명
1	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
2	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.
3	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.

11. 활성화/비활성화

11.1. QCELL

QCELL 객체에 대한 동작의 활성화/비활성화를 제어할 수 있습니다.

1) setDisable

QCELL 활성화 / 비활성화 상태로 설정합니다.

Syntax & eXample

```
setDisable(bDisable) → {boolean}
```

bDisable: disable 여부 <필수> : [boolean]

```
28 QCell1.setDisable(true);
```

See result

m-0011.html 을 실행 후 개발자 도구에서 아래 코드를 입력 후 편집기능 사용하며 true 와 false 일 때 비교하며 확인

```
> qcell.setDisable(true)
< true
> qcell.setDisable(false)
< true
```

2) getDisable

QCELL 활성화 / 비활성화 상태를 반환합니다.

(setDisable(true)와 setDisable(false)일 때의 상태를 참/거짓 여부로 반환해줍니다)

Syntax & eXample

```
getDisable() → {boolean}
```

11.2. Cell

cell 대한 동작의 활성화/비활성화를 제어할 수 있습니다.

1) setCellDisable

입력한 셀에 대하여 값으로 활성화하거나 비활성화합니다. 여러 셀을 위한 setCellsDisable API 도 제공합니다.

 Syntax & eXample

```
setCellDisable(row, col, bDisable) → {boolean}
```

srow: (비)활성화할 로우 인덱스 <필수> : [number]

scol: (비)활성화할 컬럼 인덱스 여부 <필수> : [number]

bDisable: disable 여부 <필수> : [boolean]

```
30 QCell1.setCellDisable(3, 2, true);
```

 See result

m-0011.html 을 실행 후 개발자 도구에서 아래 코드를 입력 후 편집기능 사용하며 확인

```
> qcell.setCellDisable(1,2, true)
< true
> qcell.setCellDisable(1,2, false)
< true
```

2) setCellsDisable

셀들을 활성화 / 비활성화 상태로 설정하는 API 입니다.

 Syntax & eXample

```
setCellsDisable(srow, scol, erow, ecol, bDisable) → {boolean}
```

srow : 활성화할 행 처음 인덱스 <필수> : [number]

scol: 활성화할 행 처음 인덱스 <필수> : [number]

erow : 활성화할 행 마지막 인덱스 <필수> : [number]

ecol : 활성화할 열 마지막 인덱스 <필수> : [number]

bDisable : disable 여부 <필수> : [number]

 See result

11.3. Row

row 에 대한 동작의 활성화/비활성화를 제어할 수 있습니다.

1) setRowDisable

입력한 로우에 대하여 값으로 활성화하거나 비활성화 합니다.

 Syntax & eXample

```
setRowDisable(row, bDisable, area) → {boolean}
```

row: 설정할 로우 인덱스 <필수> : [number]

bDisable: disable 여부 <필수> : [boolean]

area: 설정할 영역 ("data" | "rowheader" | "all") : [string]

```
28 QCell1.setRowDisable(3, true);
```

 See result

m-0011.html 을 실행 후 개발자 도구에서 아래 코드를 입력 후 편집기능 사용하며 확인

```
> qcell.setRowDisable(1, true)
< true
> qcell.setRowDisable(1, false)
< true
```

11.4. Col

col 에 대한 동작의 활성화/비활성화를 제어할 수 있습니다.

1) setColDisable

입력한 컬럼에 대하여 값으로 활성화하거나 비활성화 합니다.

 Syntax & eXample

```
setColDisable(col, bDisable) → {boolean}
```

col: 설정할 컬럼 인덱스 <필수> : [number]

bDisable: disable 여부 <필수> : [boolean]

```
28 QCell1.setColDisable( 2, true);
```

 See result

m-0011.html 을 실행 후 개발자 도구에서 아래 코드를 입력 후 편집기능 사용하며 확인

```
> qcell.setColDisable(1, true)
< true
> qcell.setColDisable(1, false)
< true
```

12. Pagination

그리드는 기본적으로 한 영역에 모든 데이터를 보여주며, 데이터가 많은 경우엔 상하 스크롤로 데이터를 보는 것이 불편하여 일반적인 게시판처럼 한 화면에 보여주는 건수를 정하고 나머지 데이터는 여러 페이지에 나눠 보여주는데 이러한

기능을 Pagination 이라고 합니다. Pagination 은 크게 페이지별로 데이터를 서버에 요청에서 받아와 보여주는 서버 기반이 있고 모든 데이터를 내려 받아서 페이지 설정에 맞춰서 보여주는 로컬 기반이 있습니다. 예전에는 브라우저 메모리 크기나 동작의 성능 때문에 서버 기반을 선호했지만 최근에서는 로컬 기반으로 빠른 동작과 처리를 선호하고 있으며 QCELL 도 로컬 기반 Pagination 을 지원하고 있습니다.

QCELL 객체에 Pagination 을 적용하기 위해서는 아래와 같은 정보를 생성 시 설정을 해야 합니다.

Syntax & eXample

"pagination": object

하나의 object 로 설정하거나 아래와 같이 각 항목별로 설정

pagination.pageunit: 화면에 출력한 Data 의 row 수 <필수> : [number]

pagination.unitlist: 화면에 출력 설정이 가능한 data 의 row 수의 배열 <선택> : [array]

paginaiton.url: server 와 통신할 url 주소 <선택> : [string]

pagination.loadonce: url 주소가 설정되어 있을 때 server 와 통신을 한번만 할 것인지 선택 여부 <선택> : [boolean]

pagination.totalsort: sort 기능과 같이 사용할 때 전체 데이터의 sort 설정 여부 <선택> : [boolean]

pagination.param: url 주소가 설정되어 있을 때 전달하고자 하는 파라미터 정보 <선택> : [object]

pagination.mode: 사용 방법 설정 ("simple" | "extend") <선택> : [string]

pagination.pagecount: mode 가 "extend"일 때 하단에 표시할 페이지 갯수 설정 <선택> : [number]

pagination.extendmove: move 가 "extend"일 때 하단에 이동할 페이지 입력 화면 표시 <선택> : [boolean]

```
27 "pagination" : {pageunit: 10, unitlist: [10, 20, 30]},
```

See result

m-0012.html 을 실행 후 하단에 있는 Pagination 표기를 확인

	<input type="checkbox"/>	카테고리ID	등록날짜	카테고리명	사용여부	설명	이름
1	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍
2	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍
3	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍
4	<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	N	주류에 대한 카테고리입니다.	김
5	<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y	라면제품에 대한 카테고리입니다.	홍

1 - 5 of 42

1) setPageIndex

Pagination 상황에서 현재 화면에 보여줄 페이지를 설정하는 API 입니다.

JS Syntax & eXample

```
setPageIndex(index) → {boolean}
```

index: 변경할 페이지 번호 <필수> : [number]

```
27 QCell1.setPageIndex(2);
```

See result

m-0012.html 을 실행 후 개발자 도구에서 아래 코드를 입력 후 페이지 변경을 확인해주세요.

```
> qcell.setPageIndex(2)
< true
>
```

<< < 1 / 9 > >> 5 ▾	1 - 5 of 42
<< < 2 / 9 > >> 5 ▾	6 - 10 of 42

2) getPageIndex

"현재 페이지 번호"를 반환하는 API 입니다.

JS Syntax & eXample

```
getPageIndex() → {number}
```

```
27 var nPageIndex = QCell1.getPageIndex();
```

See result

m-0012.html 을 실행 후 개발자 도구에서 아래 코드를 입력한 후 값을 확인

```
> qcell.getPageIndex()
< 2
```

3) setPageUnit

“한 페이지에 보여줄 로우의 수”를 설정하는 API 입니다.

Syntax & eXample

```
setPageUnit(index) → {boolean}
```

index: 변경할 페이지 출력 행의 수 <필수> : [number]

```
27 QCell1.setPageUnit(10);
```

See result

m-0012.html 을 실행 후 개발자 도구에서 아래 코드를 입력한 후 페이지당 10 개의 로우를 보여주는 것으로 변경되고 그에 따라 전체 페이지 수가 변하는 것을 확인해주세요.

```
> qcell.setPageUnit(10)
< true
```

<< < 1 / 9 > >> 5 ▾ 1 - 5 of 42

<< < 1 / 5 > >> 10 ▾ 1 - 10 of 42

4) getPageUnit

“한 페이지에 보여주는 로우의 수”를 반환하는 API 입니다.

Syntax & eXample

```
getPageUnit() → {number}
```

```
27 var nPageUnit = QCell1.getPageUnit();
```

See result

m-0012.html 을 실행 후 개발자 도구에서 아래 코드를 입력한 후 값을 확인해주세요.

```
> qcell.getPageUnit()
< 5
```

5) setPaginationParam

서버 기반 Pagination 을 하는 경우 서버에 원하는 페이지와 보여줄 데이터 로우 수를 요청해야 합니다. 그리고 서버는 결과(JSON 형식)와 함께 전체 건수를 내려받아야 페이지 수, 전체 데이터 건수를 화면에 보여줄 수 있습니다.

 Syntax & eXample

`setPaginationParam(property) → {boolean}`

property: 서버로 전송시 전달할 파라미터(JSON) <필수> : [object]

```
27 QCell1.setPaginationParam({"param1": 1, "param2": "RightTech"});
```

6) getPaginationParam

pagination 속성 "param"을 반환하는 API 입니다.

 Syntax & eXample

`getPaginationParam() → {Object}`

```
27 QCell1.getPaginationParam();
```

 See result

m-0012.html 을 실행 후 개발자 도구에서 아래 코드를 입력한 후 setPaginationParam 과 getPaginationParam 값을 확인해주세요.

```
> qcell.setPaginationParam({"param1": 1, "param2": "RightTech"});
< true
> qcell.getPaginationParam()
< ▼ {param1: 1, param2: "RightTech"} ⓘ
  param1: 1
  param2: "RightTech"
  ▶ __proto__: Object
>
```

7) setPaginationDatacount

pagination 속성 중 "datacount"를 설정하는 API 이며, 전체 데이터 건수의 정보를 설정합니다. 로컬 기반의 경우 전체 데이터 건수를 알 수 있지만 서버 기반의 경우엔 서버에서 알려주어야 합니다. 또한, 해당 값은 페이지 수를 계산하며 전체 페이지 정보를 계산합니다.

 Syntax & eXample

`setPaginationDatacount(count) → {boolean}`

count: pagination 사용 시 하단에 표시할 전체 데이터의 수 <필수> : [number]

```
27 QCell1.setPaginationDatacount(100);
```

8) getPaginationDatacount

pagination 속성 중 "datacount" 값을 반환하는 API 입니다.

Syntax & eXample

```
getPaginationDatacount() → {number}
```

See result

m-0012.html 을 실행 후 개발자 도구에서 아래 코드를 입력하여 확인해주세요

```
> qcell.getPaginationDatacount()
< 42
```

9) setPaginationUrl

서버 기반 Pagination 시 "해당 페이지에 대한 데이터를 요청할 url"를 설정하는 API 입니다.

Syntax & eXample

```
setPaginationUrl(url) → {boolean}
```

url: 페이지 변경시 전송할 서버 url 주소 <필수> : [string]

```
27 QCell1.setPaginationUrl("/loadurl.do");
```

10) getPaginationUrl

pagination 속성 "url"을 반환하는 API 입니다.

Syntax & eXample

```
getPaginationUrl() → {string}
```

```
27 QCell1.getPaginationUrl();
```

See result

m-0012.html 을 실행 후 개발자 도구에서 아래 코드를 입력하여 확인해주세요.

```
> qcell.setPaginationUrl("/loadurl.do");
< true
> qcell.getPaginationUrl()
< "/loadurl.do"
>
```

11) setPaginationLoadonce

pagination 속성 "loadonce"을 설정하는 API 입니다.

 Syntax & eXample

```
setPaginationLoadonce(loadonce) → {boolean}
```

bool: QCELL 생성시 서버 통신을 한번만 할 것인지 설정 <필수> : [boolean]

```
27 QCell1.setPaginationLoadonce(true);
```

12) getPaginationLoadonce

pagination 속성 " loadonce "을 반환하는 API 입니다.

 Syntax & eXample

```
getPaginationLoadonce() → {boolean}
```

```
27 QCell1.getPaginationLoadonce();
```

13) setPaginationInitSort

pagination 사용 중 설정된 sort 정보를 초기화 하는 API 입니다.

 Syntax & eXample

```
setPaginationInitSort() → {boolean}
```

```
27 QCell1.setPaginationInitSort();
```

 **check eXample** 전체 소스는 manual 내 "로컬 기반 Pagination 설정 확인" 참조

 **More info.** API 문서 내 create 영역의 pagination 부분을 참조하시기 바랍니다.

13. 선행 컬럼(rowheader)

로우헤더는 각 로우의 좌측에 있는 컬럼으로 로우에 대한 넘버링, checkbox, state(CUD) 방식을 지원합니다.

13.1. 넘버링(Numbering) 방식

순차(sequence)와 역순(reverse)으로 넘버링을 지원합니다.

 **Syntax & eXample**

"rowheader": "sequence"

"none", "sequence", "reverse", "state"(순서는 아니고 상태)

 **See result**

m-0013.html 에서 각 값을 변경하며 로우 헤더 확인

카테고리ID	카테고리ID	카테고리ID	카테고리ID
FOOD-00001	1 FOOD-00001	12 FOOD-00001	<input type="radio"/> FOOD-00001
FOOD-00002	2 FOOD-00002	11 FOOD-00002	<input type="radio"/> FOOD-00002
FOOD-00001	3 FOOD-00001	10 FOOD-00001	<input checked="" type="radio"/> df
FOOD-00002	4 FOOD-00002	9 FOOD-00002	<input type="radio"/> FOOD-00002
FOOD-00001	5 FOOD-00001	8 FOOD-00001	<input checked="" type="radio"/> FOOD-00001
FOOD-00002	6 FOOD-00002	7 FOOD-00002	<input type="radio"/> FOOD-00002
FOOD-00001	7 FOOD-00001	6 FOOD-00001	<input type="radio"/> FOOD-00001
FOOD-00002	8 FOOD-00002	5 FOOD-00002	<input type="radio"/> FOOD-00002

none sequence reverse state

13.2. checkbox 방식

넘버링 외에 로우에 대한 선택 여부에 대한 상호작용을 위해 checkbox 를 지원합니다.

1) getRowheaderChecked

rowheader 가 "checkbox" 일 때 체크된 로우들의 인덱스를 반환하는 API 입니다.

 **Syntax & eXample**

```
getRowheaderChecked(nCol) → {Array}
```

nCol: rowheader 중 checkbox 인 컬럼 index <필수> : [number]

```
27 QCell11.getRowheaderChecked(1);
```

 **See result**

m-0013.html 파일을 열어 rowheader 를 checkbox 로 설정 후 개발자 도구에서 아래 코드를 입력하여 확인해주세요

<input type="checkbox"/>	카테고리ID	등록날짜	카테고리명	사용여부
<input checked="" type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y
<input checked="" type="checkbox"/>	FOOD-00002	2019-04-10	주류	N
<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y
<input checked="" type="checkbox"/>	FOOD-00002	2019-04-10	주류	N
<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y

```
> qcell.getRowheaderChecked(0)
< ▶ (3) [1, 2, 4]
> |
```

2) setRowheaderChecked

rowheader 가 "checkbox"일 때 입력 값에 해당하는 로우와 컬럼의 인덱스의 체크 박스 값을 설정하는 API 입니다.

Syntax & eXample

`setRowheaderChecked(nRow, nCol, bState) → {boolean}`

nRow: rowheader 중 checkbox 이며 변경할 로우 index <필수> : [number]

nCol: rowheader 중 checkbox 이며 변경할 컬럼 index <필수> : [number]

bState: rowheader 중 checkbox 이며 변경할 상태 값 <필수> : [boolean]

```
27 QCell11.setRowheaderChecked(1, 0, true);
```

See result

m-0013.html 파일을 열어 rowheader 를 checkbox 로 설정 후 개발자 도구에서 아래 코드를 입력하여 확인해주세요

```
> qcell.setRowheaderChecked(1,0, true)
< true
> qcell.setRowheaderChecked(3,0, true)
< true
```

<input type="checkbox"/>	카테고리ID	등록날짜	카테고리명	사용여부
<input checked="" type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y
<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	N
<input checked="" type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y

3) setRowheaderCheckedArr

rowheader 가 "checkbox"일 때 해당 컬럼 인덱스의 여러 로우의 값을 true(체크상태)로 만들 때 사용하는 API 입니다.

 Syntax & eXample

```
setRowheaderCheckedArr(arrRow, nCol) → {boolean}
```

arrRow: rowheader 중 checkbox 이며 변경할 행 index 들의 배열 <필수> : [array]

nCol: rowheader 중 checkbox 이며 변경할 컬럼 index <필수> : [number]

```
27 QCell11.setRowheaderCheckedArr([1, 3, 5], 0);
```

 See result

m-0013.html 파일을 열어 rowheader 를 checkbox 로 설정 후 개발자 도구에서 아래 코드를 입력하여 확인해주세요.

```
> qcell.setRowheaderCheckedArr([1, 3], 0);
< true
> |
```

<input type="checkbox"/>	카테고리ID	등록날짜	카테고리명	사용여부
<input checked="" type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y
<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	N
<input checked="" type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y

 check eXample 전체 소스는 manual 내 "rowheader 에 대한 확인" 참조

4) getCheckedRows

특정 열의 타입이 "checkbox"일 때, 선택된 행 인덱스들을 배열로 반환하는 API 입니다.

 Syntax & eXample

```
getCheckedRows(col) → {Array}
```

Col: 열 인덱스 <필수> : [number]

```
qcell.getCheckedRows(2);
```

 See result

m-0013.html 파일을 열어 개발자 도구에서 아래 코드를 입력하여 확인해주세요.

```
> qcell.getCheckedRows(2)
< ▶ (2) [2, 3]
```

	<input type="checkbox"/>	<input type="checkbox"/>	OK/ng	1/0	라디오 체크
1	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>
2	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="radio"/>
3	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="radio"/>

i More info. rowheaders : ["checkbox"]로 만든 체크박스의 값은 반환 하지 못합니다.

14. Merge

머지(병합)는 이웃한 셀과 같은 값이면 하나로 묶어서 보여주는 상태를 말합니다. Merge 는 크게 헤더 영역과 데이터 영역에서 가능하며, 같은 로우를 기반으로 동작하는 기능과 같은 컬럼 기반으로 동작하는 기능이 있습니다. QCELL 객체가 만들어질 때 머지하는 것과 동적으로 머지할 수 있도록 API 를 지원합니다.

14.1. header 영역

header 영역의 merge 방식은 "none", "row", "col", "rowandcol", "colandrow"의 5 가지 방식이 있습니다.

Syntax & eXample

```
merge(strArea, strMergeType) → {boolean}
```

strArea: 병합할 영역 ("header" | "data") 중 하나 <필수> : [string]

strMergeType: header 영역의 merge 방식 중 하나 <필수> : [string]

"header" 영역은 "row" | "col" | "rowandcol" | "colandrow" | "none"

```
27 QCell1.merge("header", "rowandcol");
```

See result

m-0014.html 실행 후 개발자 도구에서 아래 코드를 입력하여 확인해주세요

<input type="checkbox"/>	지역	지역	점포명	맛 평가
<input type="checkbox"/>	서울특별시	전역	전역	평가

```
> qcell.merge("header", "row");
< true
```

<input type="checkbox"/>	지역	지역	점포명	맛 평가
<input type="checkbox"/>	서울특별시	전역	전역	평가

14.2. data 영역

data 영역의 merge 방식은 "none", "row", "col", "group"의 4 가지 방식이 있습니다.

Syntax & eXample

```
merge(strArea, strMergeType) → {boolean}
```

strArea: 병합할 영역 ("header" | "data") 중 하나 <필수> : [string]

strMergeType: header 영역의 merge 방식 중 하나 <필수> : [string]

"data" 영역은 "row" | "col" | "group" | "none"

See result

m-0014.html 실행 후 개발자 도구에서 아래 코드를 입력하여 확인해주세요

<input type="checkbox"/>	지역	지역	점포명	맛 평가
<input type="checkbox"/>	서울특별시	전역	전역	평가
<input type="checkbox"/>	서울특별시	성동구	A반점	매우 맛있음
<input type="checkbox"/>	서울특별시	성동구	B중화요리	호불호 강함
<input type="checkbox"/>	서울특별시	종로구	C차이나타운	보통
<input type="checkbox"/>	서울특별시	강남구	C차이나타운	보통
<input type="checkbox"/>	서울특별시	강남구	A반점	보통

```
> qcell.merge("data", "row");
< true
```

<input type="checkbox"/>	지역	지역	점포명	맛 평가
<input type="checkbox"/>	서울특별시	전역		평가
<input type="checkbox"/>	서울특별시	성동구	A반점	매우 맛있음

```
> qcell.merge("data", "col");
< true
```

<input type="checkbox"/>	지역	지역	점포명	맛 평가
<input type="checkbox"/>	서울특별시	전역	전역	평가
<input type="checkbox"/>		성동구	A반점	매우 맛있음
<input type="checkbox"/>			B중화요리	호불호 강함
<input type="checkbox"/>			C차이나타운	보통
<input type="checkbox"/>		강남구	A반점	

```
> qcell.merge("data", "group");
< true
```

<input type="checkbox"/>	지역	지역	점포명	맛 평가
<input type="checkbox"/>	서울특별시	전역	전역	평가
<input type="checkbox"/>		성동구	A반점	매우 맛있음
<input type="checkbox"/>			B중화요리	호불호 강함
<input type="checkbox"/>		종로구	C차이나타운	보통
<input type="checkbox"/>		강남구	C차이나타운	보통
<input type="checkbox"/>			A반점	보통
<input type="checkbox"/>				

 **check eXample** 전체 소스는 manual 내 “머지(병합) 동작에 대한 확인” 참조

15. Excel

QCELL 과 같은 웹그리드에 바인딩하여 보여지고 있는 데이터를 엑셀로 저장(export, download)하거나 엑셀 형태로 된 데이터를 QCELL 에 올려(import, upload) 데이터 처리하는 걸 지원합니다.

1) excelDownload

현재 QCELL 에 표시되고 있는 데이터를 컬럼의 정보를 포함하여 Excel 로 저장하는 API 입니다.

 **Syntax & eXample**

```
excelDownload(properties) → {boolean}
```

properties: Excel 저장 정보(JSON) **<필수> : [object]**

properties.filename: 저장할 파일명 설정 **<필수> : [string]**

properties.url: 서버 다운로드 시 호출할 URL 설정 **<필수> : [string]**

properties.border: Excel 문서에 각 영역별 border 처리 여부 설정 **<선택> : [boolean]**

properties.headershow: Excel 문서에 Header 영역 출력 여부 설정 **<선택> : [boolean]**

properties.colwidth: Excel 문서에 QCELL 과 같은 column 별 “px” 적용 여부 설정 **<선택> : [boolean]**

properties.huge: 대용량 데이터 여부 설정 **<선택> : [boolean]**

perperties.label: col 의 type 이 selectmenu 일 때 value 값이 아닌 label 으로 출력 **<선택> : [boolean]**

perperties.param: 서버로 전송할 파라미터(JSON) **<선택> : [object]**

properties.addColumnns: Excel 저장시 추가할 Column 정보 **<선택> : [object]**

properties.addColumnns.key: 추가할 Column 정보의 key **<선택> : [string]**

properties.addColumnns.width: 추가할 Column 정보의 width **<선택> : [string]**

properties.addColumns.position: 추가할 Column 정보의 위치할 컬럼 인덱스 <선택> : [string]

properties.addColumns.title: 추가할 Column 의 헤더 타이틀 <선택> : [string]

properties.delColumns: 삭제할 Column 의 index 배열 <선택> : [array]

properties.orderColumns: 순서 변경할 Column 의 index 배열 <선택> : [array]

orderColumns > delColumns > addColumns 순으로 실행

```

27 var properties =
28 {
29   filename:      "excel",
30   border:        true,
31   headershow:    true,
32   colwidth:      true,
33   param:         {"data1": "data1", "data2": "data2"},
34   addColumns:    [{"key": "abc", "width": "250", "title": ["title1", "title1-1"], "position":
35   delColumns:    [3, 4],
36   orderColumns:  [1, 6, 4, 3]
37 }
38 QCell1.excelDownload(properties);

```

 See result

m-0015.html 실행 load data 버튼을 누른 후 로컬(브라우저)기반 엑셀 저장을 해주세요.

파일 선택 선택된 파일 없음

로컬 엑셀 파일을 그리드에 업로드

로컬(브라우저)기반 엑셀 저장

load data reset grid

	시도	주차장명	유형	주소	위도	경도	관리기관
1	강원도	강동초등학교	노상	강원도 강릉시 강동면 단경길 20	37.725	128.963	강동초등학교
2		강릉초등학교			37.451		강릉초등학교
3		성덕초등학교		강원도 강릉시 용지로 60		0.000	성덕초
4		송양초등학교	노외	강원도 강릉시 성산면 위촌길 198	37.759	128.847	송양초등학교
5		신영초등학교	노상	강원도 강릉시 주문북로160	37.907	128.822	신영초등학교

	A	B	C	D	E	F	G	H
1	시도	주차장명	유형	주소	위도	경도	관리기관	
2	강원도	강동초등학교	노상	강원도 강릉시 강동면 단경길 20	37.725	128.963	강동초등학교	
3	강원도	강릉초등학교	노상		37.451		강릉초등학교	
4	강원도	성덕초등학교		강원도 강릉시 용지로 60	0	0	성덕초	
5	강원도	송양초등학교	노외	강원도 강릉시 성산면 위촌길 19	37.759	128.847	송양초등학교	
6	강원도	신영초등학교	노상	강원도 강릉시 주문북로160	37.907	128.822	신영초등학교	
7	강원도	신왕초등학교	부설	강원도 강릉시 연곡면 진고개로	37	128	신왕초등학교 행정실	
8	강원도	옥계초등학교	노상		0	0	옥계초등학교	
9	강원도	옥천초등학교		강원도 강릉시 수문길25	37.763	128.902	옥천초등학교	
10	강원도	옥천초등학교윤산분교	노상	강원도 강릉시 윤산길 79-15	37.731	128.937	옥천초등학교윤산분교장	

check eXample 전체 소스는 manual 내 "엑셀 저장 및 로딩 동작에 대한 확인" 참조

More info. API 문서 내 excelDownload 영역의 파라미터 부분을 참조하시기 바랍니다.

2) excelUpload

Excel 파일의 데이터를 업로드하여 화면에 출력하는 API 입니다.

Syntax & eXample

```
excelUpload(properties) → {boolean}
```

properties: Excel 업로드 정보 (JSON Object) <필수> : **[object]**

properties.headerrows: header 영역의 로우 카운트 설정 <필수> : **[number]**

properties.event: 파일 객체의 file change event, 로컬 업로드시 필수 <필수> : **[event]**

properties.cellname: QCELL 객체명 설정, 서버 업로드시 필수 <필수> : **[string]**

properties.fileid: 파일 객체의 id 설정, 서버업로드 시 필수 <필수> : **[string]**

properties.url: 서버 업로드 시 호출 url, 서버업로드 시 필수 <필수> : **[string]**

m-0014.html 실행 후 개발자 도구에서 아래 코드를 입력하여 확인해주세요.

See result

m-0015.html 실행 후 manual 폴더에 있는 엑셀 저장을 선택하여 로딩 후 확인

시도	주차장명	유형	주소	위도	경도	관리기관
no data						

	시도	주차장명	유형	주소	위도	경도	관리기관
1	강원도	강릉초등학교	노상	강원도 강릉시 토성로 103	37.451	128.532	강릉초등학교
2		성덕초등학교	부설	강원도 강릉시 용지로 60	0.000		성덕초
3		송양초등학교	노외	강원도 강릉시 성산면 위촌길 198	37.759	128.847	송양초등학교
4		신영초등학교	노상	강원도 강릉시 주문북로160	37.907	128.822	신영초등학교

 **check eXample** 전체 소스는 manual 내 “엑셀 저장 및 로딩 동작에 대한 확인” 참조

 **More info.** API 문서 내 uploadDownload 영역의 파라미터 부분을 참조하시기 바랍니다. 또한 엑셀뿐만 아니라 html 과 pdf 형식으로 download 할 수 있습니다. htmlDownload, pdfDownload API 를 참조하시기 바랍니다.

3) excelDownloadURLHuge

출력되는 내용을 대용량 엑셀파일로 저장하는 API 입니다. (서버를 통하여 저장)

기존 다운로드에서의 속성 border, headershow, colwidth, huge, label 들은 true 로 지정되어 동작하며 기존의 totaldata 는 제거

Syntax & eXample

```
excelDownloadURLHuge(properties) → {boolean}
```

properties: Excel Download 정보 설정 **<필수>** : [object]

properties.filename: 저장할 파일명 설정 **<필수>** : [stringq]

properties.filExt: 저장할 파일의 확장자명 설정 **<필수>** : [string] [Default] : xlsx

properties.url: 서버 다운로드 시 호출 URL 설정 (IE9 이하 및 서버 다운로드시 필수) **<필수>** : [string]

properties.param: 서버로 전송할 파라미터(JSON) **<필수>** : [object]

properties.addColumns: Excel 저장 시 추가할 Column 정보 **<필수>** : [object]

properties.addColumns.key: 추가할 Column 정보의 key **<필수>** : [string]

properties.addColumns.width: 추가할 Column 정보의 width(단위 “px”) **<필수>** : [string] [Default] : 100

properties.addColumns.position: 추가할 Column 정보의 열 인덱스 <필수> : [string]
[Default] : 마지막 열 인덱스

properties.addColumns.title: 추가할 Column 정보의 헤더 타이틀(멀티 타이틀 설정 가능, string 배열) <필수> : [object]

properties.delColumns: Excel 저장 시 삭제할 Column 의 index <필수> : [Array]

properties.orderColumns: Excel 저장 시 변경할 Column 의 index 순서 정보, QCELL 생성시점의 column 정보의 수가 동일해야하며, orderColumns > delColumns > addColumns 순으로 실행됨 <필수> : [Array]

 check eXample

```
var properties =
{
  filename:      "excel",
  fileExt:      "xlsx",
  param:        {"data1": "data1", "data2": "data2"},
  addColumns:   [{"key": "abc", "width": "250", "title": ["title1", "title1-1"], "position": "2"}, {...}],
  delColumns:   [3, 4, ...],
  orderColumns: [1, 6, 4, 3, ...]
};
qcell.excelDownloadURLHuge(properties);
```

4) exportCellInfoURLHuge

QCELL 에서 엑셀로 저장할 때 필요한 정보들을 추출하는 API 를 제공합니다.

excelDownloadURLHuge() API 에서 파라미터 filename 과 url 외 동일)

기존 다운로드에서의 속성 border, headershow, colwidth, huge, label 들은 true 로 지정되어 동작하며 기존의 totaldata 는 제거

 Syntax & eXample

```
exportCellInfoURLHuge(properties) → {Object}
```

properties: Excel 저장시 설정 정보 <필수> : [object]

properties.param: 서버로 전송할 파라미터(JSON) <필수> : [object]

properties.addColumns: Excel 저장 시 추가할 Column 정보 <필수> : [object]

properties.addColumns.key: 추가할 Column 정보의 key <필수> : [string]

properties.addColumns.width: 추가할 Column 정보의 width(단위 "px") <필수> : [string]

[Default] : 100

properties.addColumns.position: 추가할 Column 정보의 열 인덱스 <필수> : [string]

[Default] : 마지막 열 인덱스

properties.addColumns.title: 추가할 Column 정보의 헤더 타이틀(멀티 타이틀 설정 가능, string 배열) <필수> : [object]

properties.delColumns: Excel 저장 시 삭제할 Column 의 index 정보서버 업로드 시 호출 url , 서버업로드 시 필수 <필수> : [Array]

properties.orderColumns: Excel 저장 시 변경할 Column 의 index 순서 정보, QCELL 생성시점의 column 정보의 수가 동일해야하며, orderColumns > delColumns >

addColumns 순으로 실행됨 <필수> : [Array]

✓check eXample

var properties =

```
{
  param:          {"data1": "data1", "data2": "data2"},
  addColumns:     [{"key": "abc", "width": "250", "title": ["title1", "title1-1"], "position": "2"}, {...}],
  delColumns:     [3, 4, ...],
  orderColumns:   [1, 6, 4, 3, ...]
};
```

qcell.exportCellInfoURLHuge (properties);

5) excelExportInfo

QCELL 에서 엑셀로 저장할 때 필요한 정보들을 추출하는 API 를 제공합니다.

excelDownload() API 에서 파라미터 filename 과 url 외 동일)

JS Syntax & eXample

excelExportInfo(properties) → {Object}

properties: Excel 저장시 설정 정보 <필수> : [object]

properties.border: Excel 문서에 각 영역별 border 처리 여부 설정 <선택> : [boolean]

[Default] : false

properties.headershow: Excel 문서에 Header 영역 출력 여부 설정 <선택> : [boolean]

[Default] : false

properties.colwidth: Excel 문서에 QCELL 과 같은 column 별 "px" 적용 여부 설정

<선택> : [boolean] [Default] : false

properties.huge: 대용량 데이터 여부 설정 <선택> : **[boolean]** [Default] : false
 properties.label: 데이터를 value 값이 아닌 label 으로 출력,
 col 의 type 이 "static", "input", "textarea", "datepicker" 일 때 현재 화면의 표시되는
 값으로 출력 <선택> : **[boolean]** [Default] : false
 properties.param : 서버로 전송할 파라미터(JSON) <필수> : **[object]**
 properties.addColumnns: Excel 저장 시 추가할 Column 정보 <필수> : **[object]**
 properties.addColumnns.key: 추가할 Column 정보의 key <필수> : **[string]**
 properties.addColumnns.width: 추가할 Column 정보의 width(단위 "px") <필수> : **[string]**
[Default] : 100
 properties.addColumnns.position: 추가할 Column 정보의 열 인덱스 <필수> : **[string]**
[Default] : 마지막 열 인덱스
 properties.addColumnns.title: 추가할 Column 정보의 헤더 타이틀(멀티 타이틀 설정 가능,
 string 배열) <필수> : **[object]**
 properties.delColumns: Excel 저장 시 삭제할 Column 의 index <필수> : **[Array]**
 properties.orderColumns: Excel 저장 시 변경할 Column 의 index 순서 정보, QCELL
 생성시점의 column 정보의 수가 동일해야하며, orderColumns > delColumns >
 addColumns 순으로 실행됨 <필수> : **[Array]**

✓check eXample

```
var properties =

{
  border:          true,
  headershow:     true,
  colwidth:       true,
  param:          {"data1": "data1", "data2": "data2"},
  addColumns:     [{"key": "abc", "width": "250", "title": ["title1",
    "title1-1"], "position": "2"}, {...}],
  delColumns:     [3, 4, ...],
  orderColumns:   [1, 6, 4, 3, ...]
};

qcell.excelExportInfo(properties);
```

16. Group

1) setGroup

그룹핑하여 데이터를 효율적이고 효과적으로 활용하기 위해 입력받은 정보를 이용하여 데이터를 그룹화(Grouping)하는 API 입니다. 트리 그리드라고도 합니다.

Syntax & eXample

`setGroup(properties) → {boolean}`

properties: 그룹화 정보(JSON) <필수> : [object]

properties.type: 그룹화 타입("normal" | "tree") <필수> : [string]

properties.columns: 그룹화 할 컬럼들의 정보(JSON) <필수> : [array]

properties.columns[].key: 그룹화 할 컬럼의 키값 <필수> : [string]

properties.columns[].expand: "tree" 그룹화 타입 사용 시 펼침 여부 설정 <선택> : [boolean]

properties.columns[].header: 그룹화 시 표시 할 그룹 헤더 로우 정보 <선택> : [object]

properties.columns[].header.key: 그룹 헤더 로우에 데이터를 표시 할 컬럼의 키값 <선택> : [string]

properties.columns[].header.celldata: 그룹 헤더 로우에 표시 할 텍스트 데이터 배열 <선택> : [array]

properties.columns[].footer: 그룹화 시 표시 할 그룹 푸터 로우 정보 <선택> : [object]

properties.columns[].footer.key: 그룹 푸터 로우에 데이터를 표시 할 컬럼의 키값 <선택> : [string]

properties.columns[].footer.celldata: 그룹 푸터 로우에 표시 할 텍스트 데이터 배열 <선택> : [array]

```

27 QCell1.setGroup({
28   type   : "tree",
29   columns : [
30     {
31       key   : "sido",
32       expand : true,
33       footer : {
34         "sido" : ["sido: " : "groupvalue(0)", "의 Footer"],
35         "goongu" : ["을면동 갯수: ", "count('dong')"]
36       }
37     },
38     {
39       key   : "goongu",
40       expand : false,
41       footer : {
42         "sido" : ["군구: " : "groupvalue(0)", "의 Footer"],
43         "goongu" : ["을면동 갯수: ", "count('dong')"]
44       }
45     }
46   ]});

```

 See result

m-0016.html 실행 후 개발자 도구에서 아래 코드를 입력하여 확인해주세요

	기본정보						
	지역	구분	휴양림명	수용	면적	비용	
1	강원도						
2	강원도	공유림	치악산자연휴양림	700	1,700,000	일반(개인 2000, 청소...	강원도 원주시
3	강원도	공유림	동강전망자연휴양림	200	448,966	없음	강원도 경선군
4	강원도	국유림	태백고원자연휴양림	126	1,100,000	일반2000원+단체15...	강원도 태백시
5	강원도	사유림	횡성자연휴양림	300	498,263	숙박시설 이용객만 입...	강원도 횡성군
6	강원도	사유림	주천강자연휴양림	400	2,296,100	일반 : 2000원, 청소...	강원도 횡성군
14			총: 12개	평균: 33...	총면적: 12,406,078	해당사항없음	해당사항없음

check eXample 전체 소스는 manual 내 "setGroup(그룹핑과 푸터, 트리 그리드) 동작에 대한 확인" 참조

More info. API 문서 내 setGroup의 파라미터 부분을 참조하시기 바랍니다.

2) clearGroup

그룹화(Grouping)는 동적으로 실행하기에 필요에 따라 동적으로 해제하는 API 입니다.

Syntax & eXample

```
clearGroup() → {boolean}
```

```
26 QCell1.clearGroup();
```

See result

m-0016.html 실행 후 개발자 도구에서 아래 코드를 입력하여 확인해주세요.

```
> qcell.clearGroup()
< true
> |
```

3) setGrandTotal

QCELL은 입력받은 정보를 이용하여 QCELL의 데이터를 활용하여 Grand Total을 추가하는 API를 제공합니다.

Syntax & eXample

```
setGrandTotal(properties) → {boolean}
```

properties: 그룹화 정보(JSON) <필수> : [object]

properties.header: GrandTotal 에 표시 할 헤더 행 정보 <선택> : [object]
 properties.header[].key: GrandTotal 헤더 행에 데이터를 표시할 컬럼의 키 값 <선택> : [string]
 properties.header[].celldata: 그룹 헤더 행에 표시할 텍스트 데이터 배열(텍스트 or 통계함수)
 <선택> : [Array.<string>]
 properties.header[].celldata.chilcount: 직계 자식노드들의 개수 <선택> : [number]
 properties.header[].celldata.count: 전체 자식노드들의 개수 <선택>: [number]
 properties.header[].celldata.countif(colkey,value): 전체 자식노드들 중에서 지정한 컬럼의 데이터 기준값과 동일한 자식노드들의 개수 <선택>: [number]
 properties.header[].celldata.sum(colkey): 전체 자식노드 데이터들의 총합(colkey 값 미입력 시 데이터를 표시할 컬럼의 키 값으로 설정 <선택>: [number]
 properties.header[].celldata.average(colkey): 전체 자식노드 데이터들의 평균(colkey 값 미입력 시 데이터를 표시할 컬럼의 키 값으로 설정) <선택>: [number]

properties.footer: 그룹화 시 표시할 그룹 푸터 행 정보 <선택> : [object]
 properties.footer[].key: 그룹 푸터 행에 데이터를 표시할 컬럼의 키 값 <선택> : [object]
 properties.footer[] .celldata: 그룹 헤더 행에 표시할 텍스트 데이터 배열(텍스트 or 통계함수)
 <선택> : [Array.<string>]
 properties.footer[].celldata.chilcount: 직계 자식노드들의 개수 <선택> : [number]
 properties.footer[].celldata.count: 전체 자식노드들의 개수 <선택>: [number]
 properties.footer[].celldata.countif(colkey,value): 전체 자식노드들 중에서 지정한 컬럼의 데이터 기준값과 동일한 자식노드들의 개수 <선택>: [number]
 properties.footer[].celldata.sum(colkey): 전체 자식노드 데이터들의 총합(colkey 값 미입력 시 데이터를 표시할 컬럼의 키 값으로 설정 <선택>: [number]
 properties.footer[].celldata.average(colkey): 전체 자식노드 데이터들의 평균(colkey 값 미입력 시 데이터를 표시할 컬럼의 키 값으로 설정) <선택>: [number]

```
qcell.setGrandTotal({
  header : [{
    "site_name" : [" header"],
    "tqty" : [" 공수: ", "sum()", "MM"],
    "tamt" : [" 투입금액: ", "sum()", "원"]
  } ],
  footer : [{
    "site_name" : [" Footer1"],
    "tqty" : [" 공수: ", "sum()", "MM"],
    "tamt" : [" 투입금액: ", "sum()", "원"]
  }, {
    "site_name" : [" Footer2"],
    "tqty" : [" 공수 평균: ", "average()", "MM"],
    "tamt" : [" 투입금액 평균: ", "average()", "원"]
  }]
});
```

 See result

현장명	공수	투입금액
header	공수: 34.60MM	투입금액: 5,710,000원
강동구 길동 1공구현장	1.00	50,000
서초구 양재동 신축 현장	33.60	5,660,000
Footer1	공수: 34.60MM	투입금액: 5,710,000원
Footer2	공수 평균: 17.30MM	투입금액 평균: 2,855,000원

 **check eXample** 전체 소스는 manual 내 “그리드 컬럼 헤더와 푸터에 텍스트 및 통계함수 설정” 참조

17. Contextmenu

QCELL 은 생성한 그리드 내에서 사용할 기능이나 동작을 메뉴로 만들어 팝업하여 선택하는 메뉴인 컨텍스트 메뉴로 제공합니다.

1) setContextMenu

컨텍스트 메뉴를 생성하는 API 입니다. 입력 값으로 컨텍스트 메뉴 구조와 표현 방법, 선택시 동작해야 할 콜백 함수를 정의한 객체를 전달하면 생성이 되고 생성 후에는 상황에 맞게 메뉴를 보여주고 선택을 하면 QCELL 객체가 처리를 해줍니다.

 Syntax & eXample

`setContextmenu(obj) → {boolean}`

properties: 컨텍스트 메뉴 구조와 동작 정보(array) <필수> : [object]

```

30  qcell.setContextmenu(
31  {
32    header: {
33      id: "header-contextmenu",
34      contents:
35      [
36        {icon: '<i class="fas fa-plus"></i>',
37          label: '메뉴header-a',
38          callback: function(){qcell.addRow();}},
39        {divine: true},
40        {icon: '<i class="fas fa-minus"></i>',
41          label: '메뉴header-b',
42          callback: function(){qcell.removeRow();}}
43      ]
44  },

```

 See result

m-0017.html 실행 후 동작 확인 및 소스 확인



 check eXample 전체 소스는 manual 내 "컨텍스트(팝업) 메뉴 생성과 설정에 대한 확인" 참조

2) setDisableContextmenu

contextmenu 에서 비활성화할 메뉴의 id 를 설정하는 API 입니다.

 Syntax & eXample

`setDisableContextmenu(ids) → {boolean}`

ids: 비활성화할 contextmenu 의 메뉴 id 들의 배열 <필수> : [array]

```

29  QCell1.setDisableContextmenu(['header-contextmenu', 'base-contextmenu']);

```

 See result

m-0017.html 실행 후 개발자 도구에서 아래 코드를 입력 후 동작을 확인해주세요

```
> qcell.setDisableContextmenu(['header-contextmenu']);
< true
> |
```

	<input type="checkbox"/>	OK/ng	1/0	라디오 체크	firstname	lastname	age	phone	
1	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="radio"/>	Robbie	Goodman	44	970-4412-705	19
2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="radio"/>				38-4902-720	20
3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="radio"/>	Mclaughlin	Conrad	49	938-4902-720	20

3) setAbleContextmenu

contextmenu 에서 비활성화 된 메뉴 중 활성화할 메뉴 id 를 설정하는 API 입니다.

Syntax & eXample

```
setAbleContextmenu(ids) → {boolean}
```

ids: 활성화할 contextmenu 의 메뉴 id 들의 배열 <필수> : [array]

```
30 QCell11.setAbleContextmenu(['header-contextmenu', 'base-contextmenu']);
```

See result

setDisableContextmenu 예시에서 사용한 비활성화한 메뉴 ID 를 아래 명령어에 넣어 실행 후 동작을 확인해주세요.

```
> qcell.setAbleContextmenu(['header-contextmenu']);
< true
```

 **More info.** getContextmenuId() API 는 마우스 커서가 현재 위치한 곳의 contextmenu 의 id 를 반환합니다.

4) getContextmenuId

현재 마우스 커서가 위치한 Contextmenu 의 id 를 반환하는 API 입니다.

Syntax & eXample

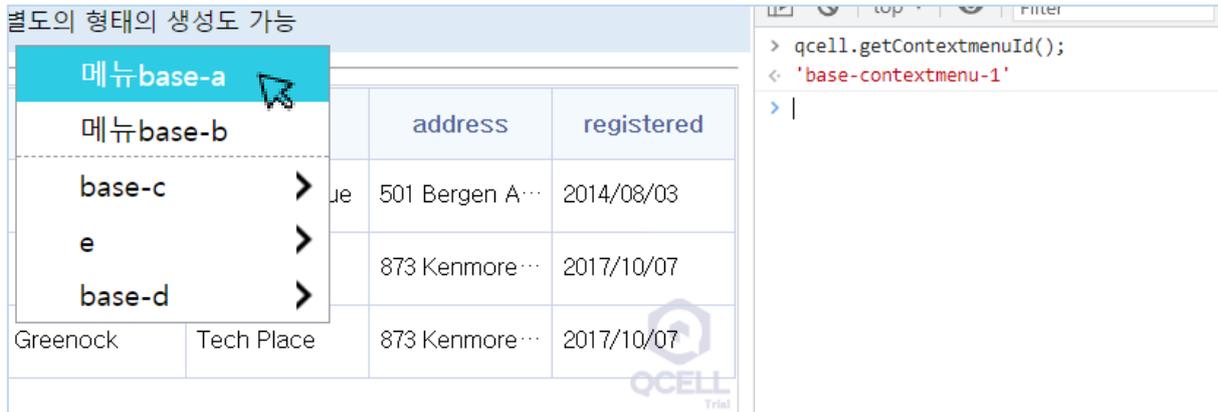
```
getContextmenuId() → {string}
```

See result

```
Ex> qcell.getContextmenuId ();
```

m-0017.html 를 열어 그리드 컬럼에서 오른쪽 마우스를 눌러 Contextmenu 를 실행합니다.

그리고 Contextmenu 의 목록에 마우스 커서를 올려 해당 명령어를 실행하면 마우스 커서가 위치한 id 를 반환합니다.



18. Event

이벤트는 가장 많이 사용되는 중요한 요소로 그리드에서 발생하는 다양한 상황(이벤트)에 맞는 동작을 수행할 수 있도록 연결해 주는 역할을 합니다.

1) bind

입력 받는 이벤트를 추가하여 처리할 수 있게 하는 API 입니다.

Syntax & eXample

```
bind(eventname, handler) → {boolean}
```

eventname: 이벤트 이름(동시에 여러 개의 이벤트 추가 시 띄어쓰기로 구분) **<필수>** : **[string]**

mousedown	QCELL UI 안에서 마우스 버튼을 눌렀을 때
mouseup	QCELL UI 안에서 마우스 버튼을 눌렀다 떼었을 때
click	QCELL UI 안에서 마우스 클릭이 발생했을 때
dblclick	QCELL UI 안에서 마우스 더블클릭이 발생했을 때

mouseenter	QCELL UI 안으로 커서가 처음 진입했을 때
mouseleave	QCELL UI 밖으로 커서가 나갔을 때
mousemove	QCELL UI 안에서 마우스 커서가 움직였을 때
keydown	QCELL UI 안에서 키보드를 눌렀을 때
valuechanged	QCELL의 셀값을 변경했을 때

handler: 추가된 이벤트 발생 시 호출할 핸들러 callback 함수 <필수> :

[requestCallback]

```

30 //event parameter에서 QCELL객체를 추출이 가능
31 QCell11.bind("click", function(event){ var objQCell = event.data.target; });
32
33 //하나의 핸들러 callback 함수를 여러개의 이벤트에 연동이 가능
34 QCell11.bind("click keydown", function(event){ var objQCell = event.data.target; });

```



m-0018.html 을 실행 후 클릭과 값 변경 후 alert 메시지 확인

<input type="checkbox"/>	카테고리ID	등록날짜	카테고리명	사용여부
<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y
<input checked="" type="checkbox"/>	FOOD-00002	2019-04-10	음료류	N
<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y
<input type="checkbox"/>	FOOD-00002	2019-04-10	주류	N
<input type="checkbox"/>	FOOD-00001	2019-04-10	라면류	Y

이 페이지 내용:
2번째 로우, 3번째 로우에서 클릭이 발생하였습니다.

[확인](#)

이 페이지 내용:
2번째 로우, 3번째 컬럼의 값이 변경되었습니다.

[확인](#)

check eXample 전체 소스는 manual 내 "이벤트 등록과 해제 동작 확인" 참조

More info. API 문서의 bind 영역에 더 많은 이벤트 정보가 있습니다.

2) unbind

입력받은 이벤트를 바인딩에서 제거하는 API 입니다. 즉, QCELL 객체가 해당 이벤트를 받지 않는 것입니다.



`unbind(eventname) → {boolean}`

eventname: 이벤트 이름(동시에 여러 개의 이벤트 제거 시 띄어쓰기로 구분) <필수> :

[string]

```

30 QCell11.unbind("click");
31 QCell11.unbind("click keydown");

```

 See result

m-0018.html 을 실행하고 개발자 도구에서 아래 코드 실행 후에 이벤트 동작 확인

```
> qcell.unbind('click', 'valuechanged');
< true
```

3) getUserEventInfo

추가된 사용자 이벤트 정보를 반환하는 API 입니다.

 Syntax & eXample

`getUserEventInfo()` → {Object}

```
30 var objUserInfo = QCell1.getUserEventInfo();
```

4) trigger

입력한 이벤트의 핸들러를 강제로 1 회 실행시키는 API 입니다.

 Syntax & eXample

`trigger(eventName)` → {boolean}

eventName: 이벤트 이름 <필수> : [string]

```
30 QCell1.trigger("click");
```

 More info. 이벤트 발생 시 전달하고 싶은 데이터를 이벤트 객체를 통해 핸들러에 아래와 같이 전달할 수 있습니다.

```
32 datagrid1.bind("valuechanged", xecuteOnValueChanged, {'gridNum':'datagrid1'});
33
34 function xecuteOnValueChanged(event) {
35     var id = event.data.gridNum; // 좌측 코드는 자바스크립트 표준 코드입니다.
36     var gdNm= QCELL.getInstance(id); // 큐셀 객체를 얻어냅니다.
37     var row = targetgrid..getIdx("row"); //값이 변경된 로우를 얻습니다. 이런 형태로 코딩을 하면 됩니다.
38     ...
39 }
```

5) clickCell

특정 cell 을 마우스로 click 한 것과 동일한 효과의 선택 동작을 하는 API 입니다. 이 기능을 사용하려면 클릭할 셀이 화면에 보이는 상태여야 합니다. 즉, 숨겨진 셀은 클릭할 수 없습니다.

 Syntax & eXample

`clickCell(row, col) → {boolean}`

row: 로우 인덱스 <필수> : [number]

col: 컬럼 인덱스 <필수> : [number]

```
30 QCell1.clickCell(1, 3);
```

 See result

m-0007.html 실행 후 개발자 도구를 띄워 아래 코드를 입력하여 확인해주세요.

```
> qcell.clickCell(2,3)
< true
> |
```

clickCell 실행 전

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

(2, 3) clickCell 실행 후

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

6) dblClickCell

특정 Cell 을 마우스로 double click 한 것과 동일한 효과로 선택하는 API 입니다. 이 기능을 사용하려면 클릭할 셀이 화면에 보이는 상태여야 합니다. 즉, 숨겨진 셀은 클릭할 수 없습니다.

 Syntax & eXample

`dblClickCell(row, col) → {boolean}`

row : 행 인덱스 <필수> : [number]

col : 열 인덱스 <필수> : [number]

```
qcell.dblClickCell(1, 3);
```

 See result

m-0007.html 실행 후 개발자 도구를 띄워 아래 코드를 입력하여 확인해주세요.

```
> qcell.dblClickCell(1,3)
< true
```

dblclickCell 실행 전

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

(1,3) dblclickCell 실행 후

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN
2	<input type="checkbox"/>	Mclaughlin		49	938-4902-720	2015/09/28	COGENTRY

7) focusCell

특정 Cell 을 focus 효과를 내며 선택하는 API 입니다.

Syntax & eXample

```
focusCell(row, col) → {boolean}
```

row : 행 인덱스 <필수> : [number]

col : 열 인덱스 <필수> : [number]

```
qcell.focusCell (2, 3);
```

See result

m-0007.html 실행 후 개발자 도구를 띄워 아래 코드를 입력하여 확인해주세요.

```
qcell.focusCell(2,3);
true
```

focusCell 실행 전

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

(2, 3) focusCell 실행 후

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

19. Extend

18.1 렌더링

QCELL 은 html 렌더링 컬럼 타입을 아래와 같이 지원합니다.

- 1) html 속성을 통해 컬럼 별 renderer 기능을 설정합니다.
- 2) html.header 속성을 통해 컬럼의 헤더 영역 셀에 적용할 renderer 콜백 함수를 설정합니다.
- 3) html.data 속성을 통해 컬럼의 데이터 영역 셀에 적용할 renderer 콜백 함수를 설정합니다.

Syntax & eXample

"html" 컬럼타입 사용시 컬럼의 각 영역(헤더, 데이터)별 renderer 정보

Name	Type	Attributes	Description
Header	function	<optional>	컬럼의 헤더 영역 renderer 콜백 함수
data	function	<optional>	컬럼의 데이터 영역 renderer 콜백 함수

html 컬럼 정의

```
31     {key: 'email', type: "html", width: '12%', title: ['연락정보', '이메일'],
32       options: {html: {data: fnEmail}}}
```

renderer 콜백 함수 정의

```
35     function fnEmail(id, row, col, val, obj){
36         var l_str=val;
37         if(val!="")
38             l_str = "<img style='width:20px;float: left; padding-top: 5px;' src='./images/email.
39                 + "<a href=mailto:"+val+">"+val+"</a>";
40         return l_str;
41     };
```

 See result

m-0009.html 을 실행하고 이메일 컬럼의 마우스 커서를 올리고 셀 클릭하여 확인

	기본정보		상세정보		연락정보				
	이름	고객번호	년간 회비	연락일	연락방법	전화번호	이메일	주소	국가
1	김철수	21-51...	₩ 1,2...	2018...	전화	82-2-345...	bancboy@ak	한국 서울 종로구 태평로 123	
2	けん たか...	21-52...	₩ 1,1...	2018...	이메일	81-3)345...	jmgomez@h	東京都港区南麻布1-2-5	

i More info. API 문서의 컬럼 타입 영역을 참조해 주세요.

18.2 서식 지정

컬럼의 데이터에 서식을 지정하는 기능(포매팅)을 지원합니다.

- 1) 바인딩된 데이터를 특정 형태(포매팅 룰)로 가공하는 역할을 합니다.
- 2) 지원하는 데이터 type 은 크게 date, string, number, custom 의 4 종류가 있습니다.

Syntax & eXample

"static", "input", "datepicker" 컬럼 타입의 "option" 설정

Name	Type	Attributes	Description
type	string	<optional>	"static", "input" 컬럼타입 사용시 포맷 타입 ("string" "number" "date" "custom")
origin	string	<optional>	"static", "input" 컬럼타입 사용시 데이터의 원본 날짜 포맷
rule	string	<optional>	"static", "input" 컬럼타입 사용시 데이터를 출력할 포맷
locale	string	<optional>	"static", "input" 컬럼타입 사용시 "type"이 "number" 나 "date"일 때 사용하려는 국가코드 입력
callback	function	<optional>	"static", "input" 컬럼타입 사용시 "custom" 포맷 타입에서만 사용이 가능하며, 커스터마이징한 format 함수

여러 서식 형태 정의

```

30     {key: 'name', type: "input", width: '150', title: ['기본정보', '이름'], styleclassname: {"data":
31       options : {format: {type: "string", rule:"@님"}}, move: true}
32     , {key: 'id', title: ['기본정보', "고객번호"], sort: true, width:'100', styleclassname: {"data
33       options : {format: {type: "custom", callback: customIden}} }
34     , {key: 'cost', type: "input",width: '100', title: ['상세정보', '년간 회비'], styleclassname:
35       options : {format: {type: "number", rule:"₩ #,###.00"}}}
36     , {key:"contact_date", type:'datepicker', title:['상세정보', '연락일'], sort: true, width:'100
37       options : {format: {type: "date", origin: "YYYYMMDD", rule: "YYYY-MM-DD"}}, stylecla

```

 See result

m-0009.html 을 실행하고 이름, 고객번호, 회비, 연락일 컬럼의 서식 적용 확인해주세요.

	기본정보		상세정보		연락정보			
	이름	고객번호	연간 회비	연락일	연락방법	전화번호	이메일	
1	김철수님	21-512-XXXX	₩ 1,200.00	2018-02-21	전화	82-2-3452-...	bancboy@ac	한국 서울
2	けん たかくら	21-522-XXXX	₩ 1,100.00	2018-03-21	이메일	81-3)3452-...	jmgomez@h	東京都港区
3	আব্দুল হামিদ	11-512-XXXX	₩ 1,200.00	2018-04-21	전화	(+880) 175-...	gamma@me	4 মাদানী এ

 More info. API 문서의 컬럼 타입의 format 영역을 참조해 주세요.

18.3 빈 데이터 안내문구(emptymessage)

QCELL 객체에 데이터가 비어있을 때, 비어있는 상태임을 알려주기 위하여 텍스트로 문구를 출력하기 위해서 속성 중 "emptymessage" 키에 출력하고자 하는 텍스트 문구를 값으로 설정하면 됩니다.

 Syntax & eXample

```
"emptymessage" : string;
```

```
29 emptymessage: "no data",
```

 See result

m-0015.html 을 실행하고 화면 중앙을 확인

시도	주차장명	유형	주소	위도	경도	관리기관
no data						

 More info. emptymessage 에 적용된 스타일 클래스(rt-qc-emptymessage)를 활용하여 이미지나 다른 문구에 스타일을 적용할 수 있습니다.

5) setAbleContextmenu

contextmenu 에서 비활성화 된 메뉴 중 활성화할 메뉴 id 를 설정하는 API 입니다.

 Syntax & eXample

```
setAbleContextmenu(ids) → {boolean}
```

ids: 활성화할 contextmenu 의 메뉴 id 들의 배열 <필수> : [array]

```
30 QCell1.setAbleContextmenu(['header-contextmenu', 'base-contextmenu']);
```

 See result

m-0017.html 실행 후 개발자 도구에서 동일한 id 를 가지고 setDisablecontextmenu 실행 확인한 다음 다시 아래 명령 코드를 입력 후 동작을 확인해주세요

```
> qcell.setAbleContextmenu(['header-contextmenu']);
< true
```

 More info. getContextmenuId() API 는 마우스 커서가 현재 위치한 곳의 contextmenu 의 id 를 반환합니다.

20. 트리(Tree)

1) setTree

입력받은 정보를 이용하여 QCELL 의 데이터를 계층화(Tree)합니다.

 Syntax & eXample

```
setTree(properties) → {boolean}
```

properties : 계층화 정보(JSON) <필수> : [object]
 properties.sort : 트리의 정렬 순서 ("none" | "asc" | "desc") <필수> : [string]
 properties.level : 레벨 정보가 들어있는 컬럼 <필수> : [string]
 properties.treeUIColumn : 트리를 보여줄 컬럼 <필수> : [string]
 properties.leafindent : 트리 단일 노드의 들여쓰기를 조절하는 값 <필수> : [number]
 properties.columns : 계층화 할 컬럼들의 정보 <필수> : [array.<object>]

통계함수

properties.columns[].childcount : 직계 자식노드들의 갯수 <필수> : [number]
 properties.columns[].count: 전체 자식노드들의 갯수<필수> : [number]
 properties.columns[].countif: 전체 자식노드들 중에서 지정한 컬럼의 데이터가 기준값과 동일한 자식노드들의 갯수<필수> : [number]
 properties.columns[].sum: 전체 자식노드 데이터들의 총합(colkey 값 미입력 시 데이터를 표시 할 컬럼의 키값으로 설정)<필수> : [number]

properties.columns[].average: 전체 자식노드 데이터들의 평균(colkey 값 미입력 시 데이터를 표시 할 컬럼의 키값으로 설정) <필수> : [number]

properties.columns[].groupvalue: 전체 자식노드 데이터들의 평균(colkey 값 미입력 시 데이터를 표시 할 컬럼의 키값으로 설정) <필수> : [number]

Properties

properties.columns[].key: 계층화 할 컬럼의 키값 <필수> : [string]

properties.columns[].expand: 펼침 여부 설정 <선택> : [boolean]

properties.columns[].header : 계층화 시 표시 할 그룹 헤더 행 정보 <필수> : [object]

properties.columns[].header{}.key: 그룹 헤더 행에 데이터를 표시 할 컬럼의 키값 <필수> : [string]

properties.columns[].header{}.celldata: 그룹 헤더 행에 데이터를 표시 할 텍스트 데이터 배열(텍스트 or 통계함수) <필수> : [array.<string>]

properties.columns[].footer: 계층화 시 표시 할 그룹 푸터 행 정보 <필수> : [object]

properties.columns[].footer{}.key: 그룹 푸터행에 데이터를 표시할 컬럼의 키값 <필수> : [string]

properties.columns[].footer{}.celldata: 그룹 푸터행에 표시할 텍스트 데이터 배열(텍스트 or 통계함수) <필수> : [array.<string>]

```

qcell.setTree({//트리 생성
  sort: 'none', // 'asc' or 'desc', 'none'
  level: 'level', // 레벨 정보가 들어가 있는 컬럼
  treeUIColumn: 'code', // 트리를 보여줄 컬럼

  columns: [
    {
      key: 'lev1', //1 depth
      expand: true //초기 로딩 시 펼침여부
    },
    {
      key: 'lev2', //2 depth
      expand: false,
    },
    {
      key: 'lev3', //3 depth
      expand: false,
    },
    {
      key: 'lev4',
      expand: false,
    },
    {
      key: 'lev5',
      expand: false,
    },
    {
      key: 'lev6',
      expand: false,
    }
  ]
});

```

 See result

manual 의 m-0023.html 파일 실행하여 아래 기능들을 사용해주시기 바랍니다.

 check eXample 전체 소스는 manual 내 "트리 그리드 생성(그룹이 아닌 tree 방식)" 참조

 More info. API 문서 내 setTree 의 파라미터 부분을 참조하시기 바랍니다.

2) insertTreeNode

해당 위치에 새로운 트리 노드를 추가합니다.

 Syntax & eXample

```
insertTreeNode(arrKey, objData) → {boolean}
```

insertTreeNode. arrKey : 해야할 키 값 배열 <필수> : [array]

insertTreeNode. objData : 해야 할 실 데이터 <필수> : [object]

```
var ret;
ret = qcell.insertTreeNode(["관리기능", "시스템관리", "제공기관 관리", "test4"],
    {
        "code": "MN000025",
        "lev1": '관리기능',
        "lev2": '시스템관리',
        "lev3": '제공기관 관리',
        "lev4": 'test4',
        "lev5": '',
        "lev6": '',
        "useYn": "Y",
        "level": 4, //해당 샘플에서 표시될 트리의 위치를 지정하는 값
        "initYn": "Y",
        "searchYn": "Y",
        "newYn": "Y",
        "saveYn": "Y",
        "delYn": "Y",
        "seq": 2,
    }, "first");
console.log(ret);
```



manual 파일 내 m-0023.html 1, 2 번 항목 확인해주세요

1. 제공기관 관리 트리의 첫번째 행에 test4 행 추가

메뉴명	메뉴코드	사용여부	메뉴 레벨	버튼기능			순서
				초기화	조회	신규	
공통코드 조회9	MN000024	Y	3	Y	Y	Y	1
공통코드 조회10	MN000024	Y	3	Y	Y	Y	1
▶ 제공기관 관리	MN000025						
test1	MN000025						
test2	MN000025						
test3	MN000025						
test5	MN000025						
test6	MN000025						

2. 메뉴관리 트리에 test5 행 추가

메뉴명	메뉴코드	사용여부	메뉴 레벨	버튼기능			순서
				초기화	조회	신규	
▶ 관리기능	MN000000	Y	1	Y	Y	Y	1
▶ 시스템관리	MN000001	Y	2	Y	Y	Y	1
▶ 메뉴관리	MN000002	Y	2	Y	Y	Y	2
test5	MN000055	Y	3	Y	Y	Y	2



전체 소스는 manual 내 "트리 그리드 생성(그룹이 아닌 tree 방식)" 참조

i More info. API 문서 내 insertTreeNode 의 파라미터 부분을 참조하시기 바랍니다.

3) removeTreeNode

해당 위치에 트리 노드의 하위를 포함하여 삭제합니다.

Syntax & eXample

```
removeTreeNode(arrKey) → {boolean}
```

removeTreeNode. arrKey : 해야할 키 값 배열 <필수> : [array]

```
qcell.removeTreeNode(["관리기능", "시스템관리", "제공기관 관리", "test1"]);
```

See result

manual 파일 내 m-0023.html 3, 4, 5 번 항목 확인해주세요.

1. 제공기관 관리 트리 첫번째 행에 test1 행 삭제

메뉴명	메뉴코드	사용여부	메뉴 레벨	버튼기능			순서	
				초기화	조회	신규		
공통코드 조회9	MN000024	Y	3	Y	Y	Y	1	
공통코드 조회10	MN000024	Y	3	Y	Y	Y	1	
제공기관 관리	MN000025	Y	3	Y	Y	Y	2	
test1								
test2	공통코드 조회9	MN000024	Y	3	Y	Y	Y	1
test3	공통코드 조회10	MN000024	Y	3	Y	Y	Y	1
test5	제공기관 관리	MN000025	Y	3	Y	Y	Y	2
test6	test2	MN000025	Y	4	Y	Y	Y	2
	test3	MN000025	Y	4	Y	Y	Y	2

2. 메뉴관리 관리 트리 첫번째 행에 test5 행 삭제(2 번 항목 누른 후 4 번 항목 진행)

메뉴명	메뉴코드	사용여부	메뉴 레벨	버튼기능			순서
				초기화	조회	신규	
관리기능	MN000000	Y	1	Y	Y	Y	1
시스템관리	MN000001	Y	2	Y	Y	Y	1
메뉴관리	MN000002	Y	2	Y	Y	Y	2
test5							

3. 시스템 관리 포함한 하위 전체 삭제

메뉴명	메뉴코드	사용여부	메뉴 레벨	버튼기능			순서
				초기화	조회	신규	
관리기능	MN0000						
▶ 시스템관리	MN0000						
▶ 관리기능	MN000000	Y	1	Y	Y	Y	1
▶ 메뉴관리	MN000002	Y	2	Y	Y	Y	2

check eXample 전체 소스는 manual 내 "트리 그리드 생성(그룹이 아닌 tree 방식)" 참조

More info. API 문서 내 removeTreeNode 의 파라미터 부분을 참조하시기 바랍니다.

4) updateTreeNode

해당 위치에 트리 노드의 하위를 포함하여 삭제합니다.

Syntax & eXample

updateTreeNode(arrKey, objData) → {boolean}

updateTreeNode. arrKey : 해야할 키 값 배열 <필수> : [array]

updateTreeNode. objData : 해야할 키 값 배열 <필수> : [object]

```
qcell.updateTreeNode(["관리기능", "시스템관리", "공통코드 조회"],
{
  "code": "MN000020",
  "lev1": '관리기능',
  "lev2": '시스템관리',
  "lev3": '데이터 변경 조회',
  "lev4": '',
  "lev5": '',
  "lev6": '',
  "useYn": "N",
  "level": 3,
  "initYn": "N",
  "searchYn": "N",
  "newYn": "N",
  "saveYn": "Y",
  "delYn": "Y",
  "seq": 2,
});
```

See result

manual 파일 내 m-0023.html 6 번 항목을 확인해주세요.

1. 시스템 관리 첫번째 행을 지정한 값으로 변경

1. 2007 기준으로 (숫자) 100보다 큰 값으로 필터링		2. 업종별 컬럼 기준으로 (문자열) 여행업과 관광숙박업만 필터링			3. 업종별 컬럼 기준의 필터 초기화(reset)			
순번	업종별	통계분류1	통계분류2	통계분류3	2007	2008	2009	2010
1	전체	사업체 수	사업체 수	전체	11716	11977	12755	14281
2	전체	사업체 수	사업체 수	없음	10714	10425	11600	13115
3	전체	사업체 수	사업체 수	100㎡ 미만	237	669	207	262
4	전체	사업체 수	사업체 수	100㎡~500㎡ 미만	211	349	338	323
5	전체	사업체 수	사업체 수	500㎡~1000㎡ 미만	141	121	146	139
6	전체	사업체 수	사업체 수	1000㎡ 이상	412	413	465	442
7	전체	사업체 수	구성비	없음	91.4	87	90.9	91.8
8	전체	사업체 수	구성비	100㎡ 미만	2	5.6	1.6	1.8
9	전체	사업체 수	구성비	100㎡~500㎡ 미만	1.8	2.9	2.6	2.3
10	전체	사업체 수	구성비	500㎡~1000㎡ 미만	1.2	1	1.1	1
11	전체	사업체 수	구성비	1000㎡ 이상	3.5	3.4	3.6	3.1
12	전체	사업체 수	평균 편의시설 면면적	소계	349	334	297	307
13	전체	종감	사업체 수	전체	0	0	0	1526

필터를 사용하기 위해선 컬럼에 filter : true 를 지정해야 합니다.

```
columns : [
  {title:["업종별"], key:"c1", width:"150", filter:true, sort:true, resize:true, type:'input'},
  {title:["통계분류1"], key:"c2", width:"150", filter:true, sort:true, resize:true, type:'input'},
  {title:["통계분류2"], key:"c3", width:"150", filter:true, sort:false, resize:true, type:'input'},
  {title:["통계분류3"], key:"c4", width:"150", filter:true, sort:false, resize:true, type:'input'},
  {title:["2007"], key:"c5", width:"100", filter:true, sort:true, resize:true, datatype:'number'},
  {title:["2008"], key:"c6", width:"100", sort:true, resize:true, datatype:'number'},
  {title:["2009"], key:"c7", width:"100", sort:true, resize:true, datatype:'number'},
  {title:["2010"], key:"c8", width:"100", sort:true, resize:true, datatype:'number'},
  {title:["2011"], key:"c9", width:"100", sort:true, resize:true, datatype:'number'},
  {title:["2012"], key:"c10", width:"100", sort:true, resize:true, datatype:'number'}
],
```

숙박업만 필터링		업종별 컬럼 기준의 필터 초기화(reset)						
통계분류3	2007	2008	2009	2010	2011	2012	2013	
	11716	11977	12755	14281	16144	17000	18000	
	10714	10425	11600	13115	14834	15500	16500	
	237	669	207	262	234	250	550	
미만	211	349	338	323	400	410	610	
미만	141	121	146	139	156	160	300	
	412	413	465	442	511	560	680	
	91.4	87	90.9	91.8	91.9	92.0	880	
	2	5.6	1.6	1.8	1.4	1.5	2.0	
미만	1.8	2.9	2.6	2.3	2.5	2.6	3.0	
미만	1.2	1	1.1	1	1	1.1	1.0	
	3.5	3.4	3.6	3.1	3.2	3.3	3.0	
	349	334	297	307	309	310	320	

그리고 해당 컬럼 옆에 필터 아이콘을 누르면 필터링을 사용할 수 있습니다.

1. 2007 기준으로 (숫자) 100보다 큰 값으로 필터링		2. 업종별 컬럼 기준으로 (문자열) 여행업과 관광숙박업만 필터링			3. 업종별 컬럼 기준의	
순번	업종별	통계분류1	통계분류2	통계분류3	2007	2008
1	전체	사업체 수	사업체 수	전체	11716	11977
2	전체	사업체 수	사업체 수	없음	10714	10425
3	전체	사업체 수	사업체 수	100㎡ 미만	237	669
4	전체	사업체 수	사업체 수	100㎡~500㎡ 미만	211	349
5	전체	사업체 수	사업체 수	500㎡~1000㎡ 미만	141	121
6	전체	사업체 수	사업체 수	1000㎡ 이상	412	413
7	전체	사업체 수	평균 편의시설 연면적	소계	349	334
8	여행업	사업체 수	사업체 수	전체	8555	8499
9	여행업	사업체 수	사업체 수	없음	8378	8279
10	여행업	사업체 수	사업체 수	100㎡ 미만	124	149
11	관광숙박업	사업체 수	사업체 수	전체	761	758
12	관광숙박업	사업체 수	사업체 수	없음	192	160
13	관광숙박업	사업체 수	사업체 수	100㎡~500㎡ 미만	115	146
14	관광숙박업	사업체 수	사업체 수	1000㎡ 이상	341	342
15	관광숙박업	사업체 수	평균 편의시설 연면적	소계	4723	4609
16	관광객이용시설업	사업체 수	사업체 수	전체	203	198
17	관광객이용시설업	사업체 수	사업체 수	없음	155	153

1 번 필터링 버튼을 누르면 지정한 100 보다 큰 값이 나오도록 필터링된 것을 확인하실 수 있습니다.

1. 2007 기준으로 (숫자) 100보다 큰 값으로 필터링		2. 업종별 컬럼 기준으로 (문자열) 여행업과 관광숙박업만		
순번	업종별	통계분류1	통계분류2	통계분류3
1	전체			전체
2	전체			없음
3	전체			100㎡ 미만
4	전체			100㎡~500㎡ 미만
5	전체			500㎡~1000㎡ 미만
6	전체			1000㎡ 이상
7	전체		평균 편의시설 연면적	소계
8	여행업			전체
9	여행업			없음
10	여행업			100㎡ 미만
11	관광숙박업			전체
12	관광숙박업	사업체 수	사업체 수	없음
13	관광숙박업	사업체 수	사업체 수	100㎡~500㎡ 미만
14	관광숙박업	사업체 수	사업체 수	1000㎡ 이상
15	관광숙박업	사업체 수	평균 편의시설 연면적	소계
16	관광객이용시설업	사업체 수	사업체 수	전체

업종별

색인목록

전체

여행업

관광숙박업

관광객이용시설업

국제회의업

카지노업

적용 **초기화**

업종별 컬럼 옆에 필터 아이콘을 클릭하여 적용을 적용 버튼을 누릅니다.

순번	업종별	통계분류1	통계분류2	통계분류3	2007	
33	여행업	증감률	구성비	100㎡~500㎡ 미만	0	0
34	여행업	증감률	구성비	500㎡~1000㎡ 미만	0	0
35	여행업	증감률	구성비	1000㎡ 이상	0	0
36	여행업	증감률	평균 편의시설 연면적	소계	0	0
37	관광숙박업	사업체 수	사업체 수	전체	761	75
38	관광숙박업	사업체 수	사업체 수	없음	192	16
39	관광숙박업	사업체 수	사업체 수	100㎡ 미만	23	25
40	관광숙박업	사업체 수	사업체 수	100㎡~500㎡ 미만	115	14
41	관광숙박업	사업체 수	사업체 수	500㎡~1000㎡ 미만	90	85
42	관광숙박업	사업체 수	사업체 수	1000㎡ 이상	341	34
43	관광숙박업	사업체 수	구성비	없음	25.2	21
44	관광숙박업	사업체 수	구성비	100㎡ 미만	3	3.3
45	관광숙박업	사업체 수	구성비	100㎡~500㎡ 미만	15.1	19
46	관광숙박업	사업체 수	구성비	500㎡~1000㎡ 미만	11.8	11
47	관광숙박업	사업체 수	구성비	1000㎡ 이상	44.8	45

그리고 난 후 2 번 필터링 버튼을 누르면 지정된 업종별 컬럼 조건인 여행업과 관광숙박업만 필터링된 것을 확인하실 수 있습니다.

 **check eXample** 전체 소스는 manual 내 “첫번째, 컬럼 필터링 방법” 참조

 **More info.** API 문서 내 filter 의 파라미터 부분을 참조하시기 바랍니다.

2) filterEx

컬럼별 필터를 설정합니다.

Syntax & eXample

```
filterEx(col, filter) → {boolean}
```

col : 열 인덱스 또는 key 값 <필수> : [number]

filter : 데이터 <필수> : [any]

```
qcell.filterEx('c2', ['증감', '사업체 수']);
```

```
qcell.filterEx('2', ['증감', '사업체 수']);
```

See result

manual 파일 내의 m-0024.html 실행해주세요.

filterEx 는 filter 처럼 열 인덱스 값을 가져올 수 있습니다.

다만 열의 키 값 또한 가져올 수 있어 아래와 같습니다.

통계분류2 - 사업체 수, 구성비

2008 200 이상

통계분류2 - 필터제거

2008 - 필터제거

계분류2	통계분류3	2007	2008	2009	2010	2011	2012
수	전체	11716	11977	12755	14281	16144	18926
수	없음	10714	10425	11600	13115	14834	16784
수	100㎡ 미만	237	669	207	262	234	557
수	100㎡~500㎡ 미만	211	349	338	323	410	619
수	500㎡~1000㎡ 미만	141	121	146	139	156	304
수	1000㎡ 이상	412	413	465	442	511	662
	없음	91.4	87	90.9	91.8	91.9	88.7
	100㎡ 미만	2	5.6	1.6	1.8	1.4	2.9
	100㎡~500㎡ 미만	1.8	2.9	2.6	2.3	2.5	3.3
	500㎡~1000㎡ 미만	1.2	1	1.1	1	1	1.6
	1000㎡ 이상	3.5	3.4	3.6	3.1	3.2	3.5
의사결 연...	소계	349	334	297	307	309	327

필터 아이콘 클릭 없이 키 값으로 조건을 지정한 필터링 기능을 사용할 수 있습니다.

계분류2	통계분류3	2007	2008	2009	2010	2011	2012
	전체	11716	11977	12755	14281	16144	18926
	없음	10714	10425	11600	13115	14834	16784
	100㎡ 미만	237	669	207	262	234	557
	100㎡~500㎡ 미만	211	349	338	323	410	619
	1000㎡ 이상	412	413	465	442	511	662
의사결 연...	소계	349	334	297	307	309	327
	전체	8555	8439	8569	9284	10725	12438
	없음	8378	8279	8435	9139	10513	12149
	전체	761	758	844	896	920	1073
	1000㎡ 이상	341	342	369	344	407	472
의사결 연...	소계	4723	4609	3553	3759	4014	4010
의사결 연...	소계	594	1002	1468	1669	2044	2656
	전체	212	237	308	363	398	485

2008 년도 200 이상의 값들만 필터링 된 결과를 확인하실 수 있습니다.

check eXample 전체 소스는 manual 내 "두번째, 컬럼 필터링 방법" 참조

More info. API 문서 내 filterEx 의 파라미터 부분을 참조하시기 바랍니다.

3) resetFilter

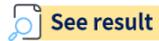
컬럼별 필터를 해제합니다.



```
resetFilter(col) → {boolean}
```

col : 열 인덱스 <필수> : [number]

```
qcell.resetFilter(1);
```



manual 파일 내의 m-0023.html 실행

1. 2007 기준으로 (숫자) 100보다 큰 값으로 필터링			2. 업종별 컬럼 기준으로 (문자열) 여행업과 관광숙박업만 필터링			3. 업종별 컬럼 기준의 필터 초기화(reset)				
순번	업종별	통계분류1	통계분류2	통계분류3	2007	2008	2009	2010	2011	2012
1	여행업	사업체 수	사업체 수	전체	8555	8499	8569	9284	10725	12438
2	여행업	사업체 수	사업체 수	없음	8378	8279	8435	9139	10513	12149
3	여행업	사업체 수	사업체 수	100㎡ 미만	124	149	79	134	78	214
4	관광숙박업	사업체 수	사업체 수	전체	761	758	844	896	920	1073
5	관광숙박업	사업체 수	사업체 수	없음	192	160	256	297	239	221
6	관광숙박업	사업체 수	사업체 수	100㎡~500㎡ 미만	115	146	127	138	160	194
7	관광숙박업	사업체 수	사업체 수	1000㎡ 이상	341	342	369	344	407	472
8	관광숙박업	사업체 수	평균 편의시설 연면적	소계	4723	4609	3553	3759	4014	4010

1 번과 2 번 필터링이 된 상태에서 3 번 초기화 버튼을 누르면 원래의 컬럼 상태로 초기화됩니다.

1. 2007 기준으로 (숫자) 100보다 큰 값으로 필터링			2. 업종별 컬럼 기준으로 (문자열) 여행업과 관광숙박업만 필터링			3. 업종별 컬럼 기준의 필터 초기화(reset)				
순번	업종별	통계분류1	통계분류2	통계분류3	2007	2008	2009	2010	2011	2012
1	전체	사업체 수	사업체 수	전체	11716	11977	12755	14281	16144	18926
2	전체	사업체 수	사업체 수	없음	10714	10425	11600	13115	14834	16784
3	전체	사업체 수	사업체 수	100㎡ 미만	237	669	207	262	234	557
4	전체	사업체 수	사업체 수	100㎡~500㎡ 미만	211	349	338	323	410	619
5	전체	사업체 수	사업체 수	500㎡~1000㎡ 미만	141	121	146	139	156	304
6	전체	사업체 수	사업체 수	1000㎡ 이상	412	413	465	442	511	662
7	전체	사업체 수	구성비	없음	91.4	87	90.9	91.8	91.9	88.7
8	전체	사업체 수	구성비	100㎡ 미만	2	5.6	1.6	1.8	1.4	2.9
9	전체	사업체 수	구성비	100㎡~500㎡ 미만	1.8	2.9	2.6	2.3	2.5	3.3
10	전체	사업체 수	구성비	500㎡~1000㎡ 미만	1.2	1	1.1	1	1	1.6
11	전체	사업체 수	구성비	1000㎡ 이상	3.5	3.4	3.6	3.1	3.2	3.5
12	전체	사업체 수	평균 편의시설 연면적	소계	349	334	297	307	309	327
13	전체	종감	사업체 수	전체	0	0	0	1526	0	0
14	전체	종감	사업체 수	없음	0	0	0	1515	0	0
15	전체	종감	사업체 수	100㎡ 미만	0	0	0	55	0	0
16	전체	종감	사업체 수	100㎡~500㎡ 미만	0	0	0	-15	0	0

✓check eXample 전체 소스는 manual 내 “첫번째, 컬럼 필터링 방법” 참조

i More info. API 문서 내 resetfilter 부분을 참조하시기 바랍니다.

4) resetFilterEx

컬럼별 필터를 해제합니다.

JS Syntax & eXample

```
resetFilterEx(col) → {boolean}
```

col : 열 인덱스 <필수> : [number]

```
qcell.resetFilterEx('c6');
```

```
qcell.resetFilterEx('6');
```

 See result

manual 파일 내의 m-0024.html 실행해주세요.

resetFilterEx 는 열 인덱스 또는 키 값을 지정하여 필터를 해제합니다.

분류2	통계분류3	2007	2008	2009	2010	2011	2012
	전체	11716	11977	12755	14281	16144	18926
	없음	10714	10425	11600	13115	14834	16784
	100m² 미만	237	669	207	262	234	557
	100m²~500m² 미만	211	349	338	323	410	619
	1000m² 이상	412	413	465	442	511	662
시설 면...	소계	349	334	297	307	309	327
	전체	8555	8499	8569	9284	10725	12438
	없음	8378	8279	8435	9139	10513	12149
	전체	761	758	844	896	920	1073
	1000m² 이상	341	342	369	344	407	472
시설 면...	소계	4723	4609	3553	3759	4014	4010
시설 면...	소계	594	1002	1468	1669	2044	2656
	전체	212	237	308	363	398	485

통계분류2 - 사업체 수, 구성비

2008 200 이상

통계분류2 - 필터제거

2008 - 필터제거

계분류2	통계분류3	2007	2008	2009	2010	2011	2012
수	전체	11716	11977	12755	14281	16144	18926
수	없음	10714	10425	11600	13115	14834	16784
수	100㎡ 미만	237	669	207	262	234	557
수	100㎡~500㎡ 미만	211	349	338	323	410	619
수	500㎡~1000㎡ 미만	141	121	146	139	156	304
수	1000㎡ 이상	412	413	465	442	511	662
	없음	91.4	87	90.9	91.8	91.9	88.7
	100㎡ 미만	2	5.6	1.6	1.8	1.4	2.9
	100㎡~500㎡ 미만	1.8	2.9	2.6	2.3	2.5	3.3
	500㎡~1000㎡ 미만	1.2	1	1.1	1	1	1.6
	1000㎡ 이상	3.5	3.4	3.6	3.1	3.2	3.5
의사결 연...	소계	349	334	297	307	309	327

check eXample 전체 소스는 manual 내 “두번째, 컬럼 필터링 방법” 참조

More info. API 문서 내 resetfilterEx 부분을 참조하시기 바랍니다.

22. 엑셀 외 저장 방식

22.1. html

1) htmlDownload

Export 시 HTML 파일로 저장하는 API 입니다.

Syntax & eXample

```
htmlDownload(properties) → {Boolean}
```

QCELL 객체의 ID. htmlDownload(properties)

properties : HTML 저장 시 설정 정보 **<필수>** : **[object]**

properties.filename : 저장할 HTML 파일명 설정 (필수) **<필수>** : **[string]**

properties.label : 데이터를 value 값이 아닌 label 으로 출력 **<필수>** : **[boolean]**

```
Ex>
var properties =
{
  filename:      "html",
  label:         true,
};
qcell.htmlDownload(properties);
```

 **check eXample** manual 내 m-0031.html 파일, 목차 목록의 “엑셀 외 저장 방식” 샘플 참조

 **More info.** manual 폴더의 m-0031.html 파일을 참조하시길 바랍니다.

22.2. txt

1) txtDownload

Export 시 txt 파일로 저장하는 API 입니다.

 **Syntax & eXample**

`txtDownload(properties) → {Boolean}`

QCELL 객체의 ID. txtDownload(properties)

properties : txt 저장 시 설정 정보 <필수> : **[object]**

properties.usehtmltype : true 면 html 값 그대로 표시(false 면 그리드 데이터 그대로 표시) <필수> : **[boolean] [Default] : false**

properties.footer : 설정된 하단 패널의 값을 표시(QCell1.getCustomPanel() API 사용 필수) <필수> : **[object]**

properties.filename : 저장할 txt 파일명 설정 (필수) <필수> : **[string]**

properties.label : 데이터를 value 값이 아닌 label 로 출력 <필수> : **[boolean] [Default] : false**

properties.headershow: Export 시 header 포함 여부 <필수> : **[boolean] [Default] : false**

properties.colsep : Export 시 Col 간 구분자 <필수> : **[string] [Default] : ‘,’**

properties.rowsep: Export 시 Row 간 구분자 <필수> : **[string] [Default] : ‘\n’**

properties.addColumns: Export 시 숨겨진 데이터 추가 할 수 있는 기능(단, excel export addColumns spec 에 있는 width 는 무시함) <필수> : **[string]**

```
Ex>
var properties =
{
  filename:      "txt",
  headershow:    true,
  addColumns:    [{"key": "abc", "width": "250", "title": ["title1",
                                                         "title1-1"], "position": "2"}, {...}],

  usehtmltype:  true,
  colsep:       true,
  rowsep:       true,
  label:        true,
  footer:       QCell1.getCustomPanel()

}
QCell1.txtDownload(properties);
```

 [check eXample](#) manual 내 m-0031.html 파일, 목차 목록의 "엑셀 외 저장 방식" 샘플 참조

22. 헤더(Header)

23.1. 제목(title) 동적 변경

1) setHeaderCellValue

header 의 title 을 동적으로 변경 시 사용합니다.

 **Syntax & eXample**

```
setHeaderCellValue(row, col, data) → {boolean}
```

row : 셀의 행 인덱스 <필수> : [number]

col : 셀의 열 인덱스 <필수> : [number]

data : 셀 데이터 <필수> : [*]

```
qcell.setHeaderCellValue(0,1,"hi")
```

 **See result**

manual 파일 내의 m-0024.html 실행

컬럼1	컬럼2	컬럼3
Hello	큐셀!	가장 간단한 큐셀 그리드입니다.



[<< 목록으로 가기](#)

개발자 도구를 열어 아래의 코드를 입력하여 주세요

```
qcell.setHeaderCellValue(0,1,"HelloWorld")
```

컬럼 1	HelloWorld	컬럼 3
Hello	큐셀!	가장 간단한 큐셀 그리드입니다.

2) setHeaderCellValueEx

header 의 title 을 동적으로 변경 시 사용합니다.

JS Syntax & eXample

```
setHeaderCellValueEx(row, key, data) → {boolean}
```

row : 셀의 행 인덱스 <필수> : [number]

key : 셀의 열 키 인덱스 <필수> : [string]

data : 셀 데이터 <필수> : [*]

```
qcell.setHeaderCellValueEx(0, 'message', "ChangeTitle")
```

See result

manual 파일 내의 m-0024.html 실행해주세요.

컬럼 1	컬럼 2	컬럼 3
Hello	큐셀!	가장 간단한 큐셀 그리드입니다.

<< 목록으로 가기

개발자 도구를 열어 아래의 코드를 입력해주세요.

```
qcell.setHeaderCellValueEx(0, 'message', "ChangeTitle")
```

컬럼 1	ChangeTitle	컬럼 3
Hello	큐셀!	가장 간단한 큐셀 그리드입니다.

i More info. 컬럼의 key 값을 입력하여 지정합니다.

3) getHeaderCellValueEx

QCELL 은 header 의 title 을 값을 반환합니다.

JS Syntax & eXample

```
getHeaderCellValueEx(row, key) → {string}
```

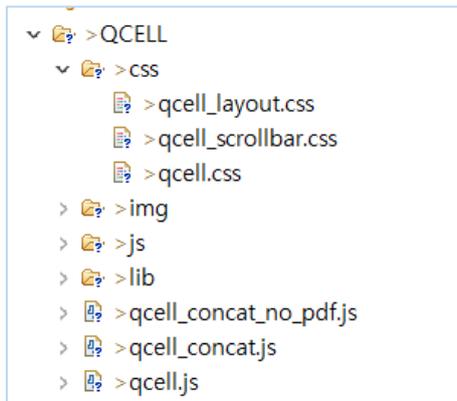
row : 셀의 행 인덱스 <필수> : [number]

key : 셀의 열 키 인덱스 <필수> : [string]

6 장. 스타일(CSS) 수정하기

1. CSS 경로와 파일

QCELL 객체가 생성될 때 미리 정의된 스타일 시트를 활용합니다. 아래 그림은 QCELL의 CSS 파일이 위치한 경로와 파일들입니다. QCELL의 내부의 구성요소에 대한 내용을 담고 있는 것이 qcell_layout.css이며, 구조 위에 그려지는 모든 대상(색상이나, 폰트 등)을 정의하고 있는 것이 qcell.css이며 개발하는 웹 시스템의 스타일에 맞춰 수정하면 됩니다. 마지막으로 데이터 건수가 많거나 컬럼이 많아서 생성되는 스크롤바에 대한 정의입니다. 이 부분도 원하는 모습으로 수정할 수 있습니다.



2. QCELL 의 구성

2.1 Header

	<input type="checkbox"/>	firstname	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Aver
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
9	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
10	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
11	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
13	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

QCELL 의 Header 영역은 colheader, rowheader, edgeheader 의 3 개 header 로 구성되어 있습니다.

1) colheader

일반적인 그리드에서 header 로 부르는 영역으로 각 col(열)의 title(제목)을 표시하는 수행하는 역할을 합니다. 이 영역에 해당하는 row(행)들을 headerrows 로 표현합니다.

CSS class

```
.rt-qc-cell-colheader, .rt-qc-cell-headerrows
```

2) rowheader

설정 에 따라 표시 가 되지 않을 수도 있지만, 대개 row(행)에 대한 numbering 을 수행하며, numbering 은 sequence(순차), reverse(역순)의 2 가지 형태를 지원합니다. 그 외의 checkbox, state 형태를 지원합니다. 이 영역에 해당하는 col(열)들을 headercols 로 표현합니다.

CSS class

```
.rt-qc-cell-rowheader, .rt-qc-cell-headercolscolheader
```

3) edgeheader

colheader 와 rowheader 의 사이에 존재하는 header 로써, colheader 와 rowheader 를 구분 짓기 위해 존재하는 header 이지만, rowheader 의 속성 중 checkbox 를 이용하게 될 때 전체 선택 checkbox 를 표기하기도 합니다. 이 영역은 headercols, headerrows 양쪽에 포함되어 있습니다.

CSS class

.rt-qc-cell-edgeheader, .rt-qc-cell-headerrows, .rt-qc-cell-headercols

2.2 Viewport(Data)

Viewport(Data)영역은 실제 data 가 표현되는 영역에 대한 명칭입니다.

viewport

	<input type="checkbox"/>	firstname	lastname	age	phone	birthday	company	city	street
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert Ave
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
9	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
10	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
11	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place
13	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Place

CSS class

.rt-qc-cell-viewport

선택은 다중과 단일로 선택할 수 있으며, 선택된 것에 대해 select class 가, 마지막으로 선택된 것에는 select class 외에 focus class 가 추가됩니다.

CSS class select

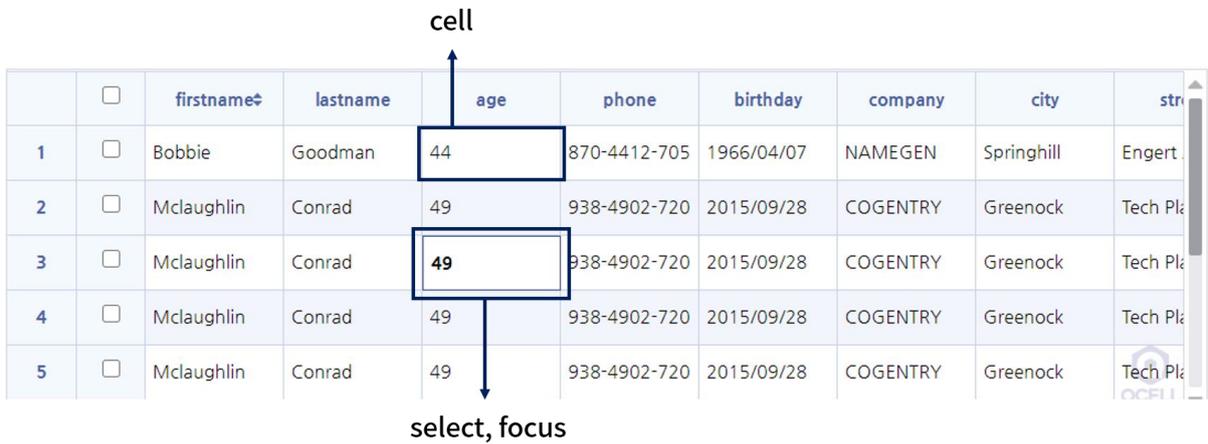
.rt-qc-cell-select

CSS class focus

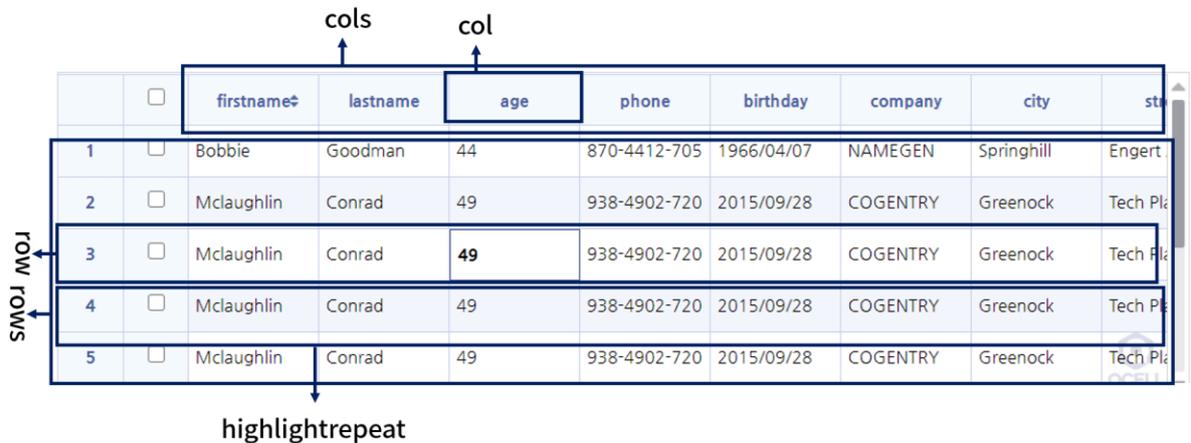
.rt-qc-cell-focus

1) cell

cell 이란 QCELL 에서 data 를 표현하는 최소 단위(칸)의 명칭입니다.



2) col, row



col(컬럼, 열) 이란 QCELL 에서 cell 의 세로 묶음의 단위의 명칭입니다.

cols 란 headercols 를 포함한 전체 col(컬럼, 열)들의 명칭입니다.

row(로우, 행) 이란 QCELL 에서 cell 의 가로 묶음의 단위의 명칭입니다.

rows 란 headerrows 를 포함한 전체 row(로우, 행)들의 명칭입니다.

col(컬럼)과 row(로우)는 qcell.css 를 통해서는 스타일을 변경할 수 없습니다.

다만, col(컬럼)은 columns 의 속성 중 styleclassname 을 통해 스타일을 적용할 수 있습니다.

row(로우)는 특정 주기로 반복되는 row(로우)에 대한 highlightrepeat 라는 속성을 설정함으로써 highlightrepeat 에 스타일을 적용할 수 있습니다.

CSS class highlightrepeat

```
.rt-qc-cell-highlightrepeat
```

3) frozencol, frozenrow, frozenbottomrow

	<input type="checkbox"/>	firstname↕	lastname	age	phone	birthday	company	city	str
1	<input type="checkbox"/>	Bobbie	Goodman	44	870-4412-705	1966/04/07	NAMEGEN	Springhill	Engert
2	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Pla
3	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Pla
4	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Pla
10	<input type="checkbox"/>	Mclaughlin	Conrad	49	938-4902-720	2015/09/28	COGENTRY	Greenock	Tech Pla

frozencols 란 QCELL 좌측에 고정되는 column(컬럼)에 대한 속성의 명칭입니다.

CSS class frozencols

```
.rt-qc-cell-frozencols
```

frozenrows 란 QCELL 상단에 고정되는 row(로우)에 대한 속성의 명칭입니다.

CSS class frozenrows

```
.rt-qc-cell-frozenrows
```

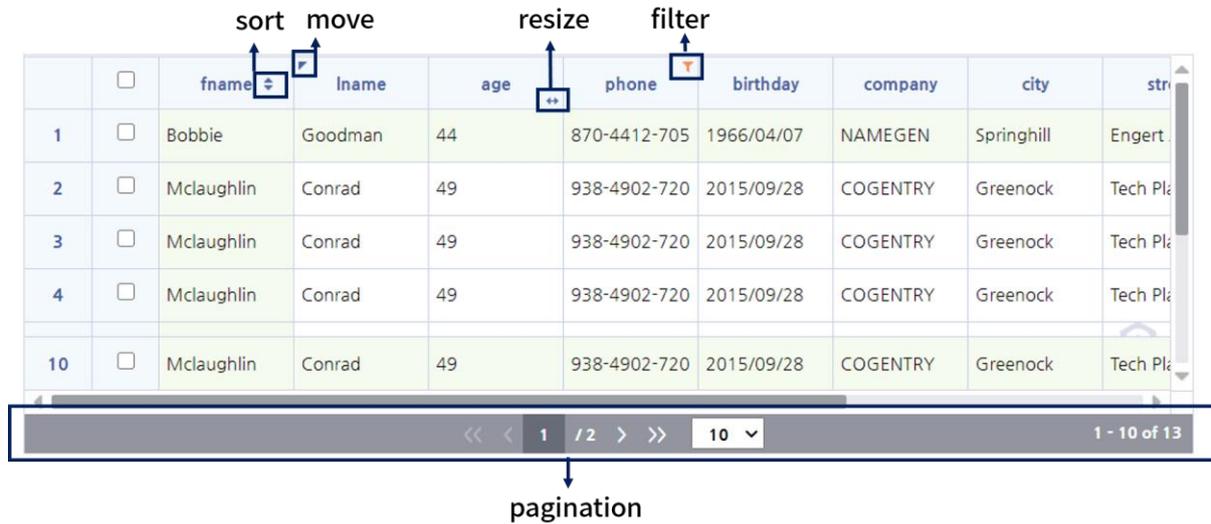
frozenbottomrows 란 QCELL 하단에 고정되는 row(로우)에 대한 속성의 명칭입니다.

CSS class frozenbottomrows

```
.rt-qc-cell-frozenbottomrows
```

2.3 기타

기타 정렬, 이동, 크기변경, 필터링 기능 동작을 사용자에게 보이도록 제공하는 스타일과 대량의 데이터일 때 pagination 스타일이 있습니다.



1) move

move 란 각 col(열)의 header 좌측 상단에 표시되는 move icon 을 drag 하여 col(열)의 순서를 바꾸는 기능을 뜻합니다.

CSS class

```
.rt-qc-colheaderdndarea
```

2) sort

sort 란 각 col(열)의 header 우측 중단에 표시되는 sort icon 을 click 하여 해당 col(열)의 data 를 오름차순, 내림차순으로 정렬하는 기능을 뜻합니다.

CSS class

```
.rt-qc-sortarea, .rt-qc-sort-asc, .rt-qc-sort-desc
```

3) resize

resize 란 각 col(열)의 header 의 우측 하단에 표시되는 resize icon 을 drag 하여 col(열)의 width 를 조정하는 기능을 뜻합니다.

CSS class

```
.rt-qc-colresizearea
```

4) pagination

pagination 이란 하나의 QCELL 에 너무 많은 data 나 특정 수만큼의 data 만 표출하려고 할 때 해당 data 정보들을 담은 page 정보들을 QCELL 하단에 표현하는 기능을 뜻합니다.

CSS class

```
.rt-qc-pagination-nav, .rt-qc-pagination-nav-btn, .rt-qc-pagination-ui-center .rt-qc-  
pagination-nav-input, .rt-qc-pagination-nav-select. .rt-qc-pagination-nav-info
```

문의.

라이트텍 홈페이지: <http://righttech.co.kr>

QCELL 기술사이트: <http://support.qcell.kr>

service@righttech.co.kr

02-2135-2734

Copyright 2020. RightTech, Inc. All right reserved

본 문서의 저작권은 주식회사 라이트텍(RightTech, Inc.)에 있습니다. 본 문서는 QCELL의 정식 라이선스를 구매하신 고객께 제공되는 문서이며, 열람 권한은 구매 고객에게 있습니다. 어떤 형태(E-Mail, 메신저, 복사본 등을 포함한 문서의 내용을 열람 또는 구두 전달을 포함)로든 무단 복사 및 수정, 라이선스를 보유하지 못한 타인 및 타 기관 전달을 엄격히 금지하며, 이를 위반할 시 법적인 책임을 물을 수 있습니다.