



Komponenten Documentation

UICA-Widgets

Inhaltsverzeichnis

a.w.trait.dataView	3
a.w.abstract.table	4
a.widget.table	15

app.widget.trait.dataView

Wrapper for SlickGrids DataView Plugin

app.widget.abstract.table

Dies ist eine abstrakte Komponente, welche Funktionalitäten zur Anbindung des table-widgets anbietet. Wird diese abstrakte Komponente eingebunden, muss sich nicht händisch um die Integration des table-widgets gekümmert werden. Überall wo eine Tabelle eingebunden wird sollte es über diese abstrakte Komponente laufen. Die Komponente stellt Methoden für DataView bereit, um einzelne Tabelleneinträge zu verändern, löschen oder hinzuzufügen. Außerdem bietet diese Komponente Methoden, um aus Daten-Objekten Präsentation-Objekte, welche die Tabelle benötigt, zu generieren. Die Komponente table wird initialisiert und die notwendigen Daten übergeben (Daten, Spalten, Optionen) Die hiermit eingebundene Tabelle ist standardmäßig - nicht multiselect über eigene Spalte (multiSelect: false) - nicht singleselect über eigene Spalte (activateSelectPlugIn: false) - aber selektierbar durch Klick irgendwo in eine Zeile (activateRowSelectionModel: true) - nicht editierbar (editable: false) - erlaubt es nicht Spalten zu verschieben (enableColumnReorder: false) - hat eine Zeilenhöhe von 27px (rowHeight: 27) - passt die Spalten auf die verfügbare Breite an (forceFitColumns: true)

API

API Definitions
api.call.table:resize
api.call.table:data
api.call.table:groupItemMetadataProvider
api.call.table:multiSelectPluginOptions
api.call.table:singleSelectPluginOptions
api.call.table:changeRowCount
api.call.table:changedRows
api.call.table:columns
api.call.table:options
api.call.table:setOnBeforeEditCellCallback
api.call.table:setOnBeforeMoveRowsCallback
api.call.table:render
api.call.table:scrollToIndex
api.call.table:selectedItems
api.register.table:changeCollapsedStates
api.register.table:invalidate

EVENTS

Events Definitions
events.publish.toParent.dataView
events.subscribe.forChildren.table:activeCellChanged
events.subscribe.forChildren.table:sortData
events.subscribe.forChildren.table:selectedDataItemsChanged

MODEL

Model Definitions

model.define.own.data.tableOptions
model.define.own.data.tableColumns
model.define.own.data.allAvailableColumns
model.define.own.data.sortObject
model.define.own.data.filterValue
model.define.own.data.activeCellsPresentationEntry
model.define.own.data.selectedTableEntriesFromTable
model.define.own.data.selectedTableEntriesFromOutside
model.observe.global.command.resize
model.observe.own.data.sortObject
model.observe.own.data.filterValue
model.observe.own.data.tableColumns
model.observe.own.data.tableOptions
model.observe.own.data.activeCellsPresentationEntry
model.observe.own.data.selectedTableEntriesFromOutside
model.observe.own.data.selectedTableEntriesFromTable

Details

api.call.table:resize

Name: table:resize
Part: widget-table

Sagt dem Tabellenwidget, dass sich das Grid in der Größe geändert hat und somit neu gerendert werden muss

Target-Component	Name
widget-table	api.register.table:resize

api.call.table:data

Name: table:data
Parameter: SlickGrid DataView Objekt
Part: widget-table
 Übergibt den Tabellenwidget sein SlickGrid DataView Objekt

Target-Component	Name
widget-table	api.register.table:data

api.call.table:groupItemMetadataProvider

Name: table:groupItemMetadataProvider
Parameter: SlickGrid Plugin
Part: widget-table
Übergibt den Tabellenwidget das Grouping-Plugin

Target-Component	Name
widget-table	api.register.table:groupItemMetadataProvider

api.call.table:multiSelectPluginOptions

Name: table:multiSelectPluginOptions
Parameter: Objekt
Part: widget-table

Target-Component	Name
widget-table	api.register.table:multiSelectPluginOptions

api.call.table:singleSelectPluginOptions

Name: table:singleSelectPluginOptions
Parameter: Objekt
Part: widget-table

Target-Component	Name
widget-table	api.register.table:singleSelectPluginOptions

api.call.table:changeRowCount

Name: table:changeRowCount
Part: widget-table
Gibt der Tabelle die Info, dass sich der Row Count geändert hat

Target-Component	Name
widget-table	api.register.table:changeRowCount

api.call.table:changedRows

Name: table:changedRows
Parameter: object
Part: widget-table
Gibt der Tabelle die Info, dass sich die Zeilen geändert haben

Target-Component	Name
widget-table	api.register.table:changedRows

api.call.table:columns

Name: table:columns
Parameter: object - Spaltendefinitions-Objekte
Part: widget-table
 Übergibt dem Tabellenwidget die Spaltendefinitionen

Target-Component	Name
widget-table	api.register.table:columns

api.call.table:options

Name: table:options
Parameter: entweder ein komplettes Objekt mit allen Optionen oder zwei Parameter: ein bestimmtes Attribut mit einem bestimmten Wert
Part: widget-table
 Übergibt dem Tabellenwidget die Optionen

Target-Component	Name
widget-table	api.register.table:options

api.call.table:setOnBeforeEditCellCallback

Name: table:setOnBeforeEditCellCallback
Parameter: function - Eine Funktion die vom Widget aus aufgerufen wird um zu überprüfen ob eine bestimmte Zelle editierbar ist
Part: widget-table
 Dieser Call darf erst im Lebenszyklus 'render' gemacht werden.

Target-Component	Name
widget-table	api.register.table:setOnBeforeEditCellCallback

api.call.table:setOnBeforeMoveRowsCallback

Name: table:setOnBeforeMoveRowsCallback
Parameter: function - Eine Funktion die vom Widget aus aufgerufen wird um zu überprüfen ob eine bestimmte Zeile verschoben werden darf.
Part: widget-table
 Dieser Call darf erst im Lebenszyklus 'render' gemacht werden.

Target-Component	Name
widget-table	api.register.table:setOnBeforeMoveRowsCallback

api.call.table:render

Name: table:render
Part: widget-table
 Sagt dem Tabellenwidget, dass das Grid gerendert werden soll.

Target-Component	Name
widget-table	api.register.table:render

api.call.table:scrollToIndex

Name: table:scrollToIndex
Part: widget-table
 Sagt dem Tabellenwidget, dass das Grid an einen bestimmten Index scrollen soll.

Target-Component	Name
widget-table	api.register.table:scrollToIndex

api.call.table:selectedItems

Name: table:selectedItems
Parameter: Array mit selektierten Zeilen (Präsentation-Objekten)
Part: table
 Übergebe die gewünschten selektierten Präsentation-Objekten der Tabelle

Target-Component	Name
widget-table	api.register.table:selectedItems

api.register.table:changeCollapsedStates

Name: table:changeCollapsedStates
Parameter: boolean
 Passt den Klappzustand aller Items auf den gegebenen Wert an.

api.register.table:invalidate

Name: table:invalidate
 Ruft den Call 'table:resize' der Tabellenkomponente auf

events.publish.toParent.dataView

Name: dataView

Überprüft ob die Parent-Komponente bereits einen DataView registriert hat, falls ja wird dieses in den dynamics 'this.dataView' gesetzt Rückgabeparamter: DataView-Objekt oder null

events.subscribe.forChildren.table:activeCellChanged

Name: table:activeCellChanged

Parameter: object - activeCellObj - das Objekt der aktiven Zelle mit dem Attribut dataObject, welche das Präsentation-Objekt darstellt.

Dieses Event wird gefangen, wenn sich die aktive Zelle in der Tabelle geändert hat. Das Modelfeld 'data:activeCellsPresentationEntry' wird auf das Präsentation-Objekt der Zeile, in der die Zelle liegt, gesetzt

Target-Component	Name
widget-table	events.publish.toParent.table:activeCellChanged

events.subscribe.forChildren.table:sortData

Name: table:sortData

Parameter: Objekt mit den Attributen 'col': Die Spalte, anhand der sortiert werden soll 'asc': 1 oder -1 - die Richtung in die sortiert werden soll

Das Modelfeld 'data:sortObject' wird befüllt mit den gegebenen Parameter angereichert mit dem Attribut 'field' = col.field

Target-Component	Name
widget-table	events.publish.toParent.table:sortData

events.subscribe.forChildren.table:selectedDataItemsChanged

Name: table:selectedDataItemsChanged

Parameter: Array von selektierten Tabelleneinträgen (Präsentation-Objekten)

Event, das gefangen wird, wenn sich die Selektion der Zeilen von der Tabelle aus geändert hat. Nur wenn genau ein Präsentation-Objekten selektiert ist wird es in das Modelfeld 'data:selectedTableEntriesFromTable' gesetzt. Ansonsten wird das Modelfeld 'data:selectedTableEntriesFromTable' auf ein leeres Array gesetzt

Target-Component	Name
widget-table	events.publish.toParent.table:selectedDataItemsChanged

model.define.own.data.tableOptions

Name: tableOptions
Parameter: Objekt mit Optionen
 Hält alle Optionen für die Tabelle

Target-Component	Name
appWidgetAbstractTable	model.observe.own.data.tableOptions

model.define.own.data.tableColumns

Name: tableColumns
Parameter: Array von Spalten-Objekten
 Hält die sichtbaren Spalten der Tabelle

Target-Component	Name
appWidgetAbstractTable	model.observe.own.data.tableColumns

model.define.own.data.allAvailableColumns

Name: allAvailableColumns
Parameter: Array von Spalten-Objekten
 Hält alle verfügbaren Spalten der Tabelle

model.define.own.data.sortObject

Name: sortObject
Parameter: Objekt mit den Attributen asc: 1 oder -1 für auf- oder absteigend field: das Attribut des Tabelleneintrags nach dem sortiert werden soll col: optional - initial nicht gefüllt - gefüllt wenn das Objekt über Klick auf Header sortiert wird und nicht wenn der initiale Sort-Wert gesetzt wird
 Hält das aktuelle Sort-Objekt

Target-Component	Name
appWidgetAbstractTable	model.observe.own.data.sortObject

model.define.own.data.filterValue

Name: filterValue
Parameter: string
 Hält den aktuellen Suchstring

Target-Component	Name
appWidgetAbstractTable	model.observe.own.data.filterValue

model.define.own.data.activeCellsPresentationEntry

Name: activeCellsPresentationEntry
Parameter: object - das Präsentation-Objekt = Zeile, von welchen gerade eine Zelle aktiv ist
 Hält von der aktiven Zelle von der dazugehörenden Zeile das Präsentation-Objekt

Target-Component	Name
appWidgetAbstractTable	model.observe.own.data.activeCellsPresentationEntry

model.define.own.data.selectedTableEntriesFromTable

Name: selectedTableEntriesFromTable
Parameter: Array von selektierten Tabelleneinträgen, welche vom der Tabelle getriggert wurden, dass sie selektiert wurden
 Hält alle selektierten Tabelleneinträgen, die von der Tabelle kommen

Target-Component	Name
appWidgetAbstractTable	model.observe.own.data.selectedTableEntriesFromTable

model.define.own.data.selectedTableEntriesFromOutside

Name: selectedTableEntriesFromOutside
Parameter: Array von selektierten Tabelleneinträgen, welche von außen in die Tabelle gesetzt werden
 Hält alle selektierten Tabelleneinträgen, die von außen gesetzt werden

Target-Component	Name
appWidgetAbstractTable	model.observe.own.data.selectedTableEntriesFromOutside

model.observe.global.command.resize

Name: resize

Der Tabelle wird gesagt sich zu resizen, es wird ein Call auf 'table:resize' aufgerufen.

model.observe.own.data.sortObject

Name: sortObject
Parameter: Objekt mit den Attributen asc: 1 oder -1 für auf- oder absteigend field: das Attribut des Tabelleneintrags nach dem sortiert werden soll col: optional - initial nicht gefüllt - gefüllt wenn das Objekt über Klick auf Header sortiert wird und nicht wenn der initiale Sort-Wert gesetzt wird
Part: Controller
Handelt es sich um einen Baum, werden die Items sortiert in die DataView gesetzt, handelt es sich um eine normale Tabelle werden die Items der DataView sortiert. Die Sortierung findet in beiden Fällen anhand der definierten Sort-Methode statt.

Target-Component	Name
appWidgetAbstractTable	model.define.own.data.sortObject

model.observe.own.data.filterValue

Name: filterValue
Parameter: string
Part: Controller
Starten die Suche in den Tabellendaten. Es werden die Zeilen herausgefiltert, die nicht den Suchstring, bzw. nicht jedes Wort des Suchstrings in der kompletten Zeile enthalten. Die einzelne Wörter können dabei auf verschiedene Zellen verteilt sein.

Target-Component	Name
appWidgetAbstractTable	model.define.own.data.filterValue

model.observe.own.data.tableColumns

Name: tableColumns
Parameter: Array mit Spalten-Objekten
Part: Controller
Die übergeben Spalten werden dem Tabellen-Widget übergeben (Call 'table:options')

Target-Component	Name
appWidgetAbstractTable	model.define.own.data.tableColumns

model.observe.own.data.tableOptions

Name: tableOptions

Parameter: ev - ComponentJS-Event newValue - Objekt (Optionen-Objekt) oder der neue Wert eines bestimmten Attributs oldValue - vorherige gesetzte Objekt (Optionen-Objekt) oder der vorherige Wert eines bestimmten Attributs op: Operation (String) - wird nicht benötigt, nicht ausgelesen path - string, der Pfad des Modelfeldes

Part: Controller

Es wird überprüft ob sich die Gesamtmenge der Optionen verändert hat oder nur eine bestimmte Option. Im ersten Fall wird ein Call auf 'table:options' gemacht und die Gesamtmenge der Optionen übergeben. Im zweiten Fall wird ein Call auf 'table:options' gemacht und das Attribut das sich geändert hat und der neue Wert übergeben.

Target-Component	Name
appWidgetAbstractTable	model.define.own.data.tableOptions

model.observe.own.data.activeCellsPresentationEntry

Name: activeCellsPresentationEntry

Parameter: ev - ComponentJS-Event newValue - das Präsentation-Objekt der aktuell aktiven Zelle oldValue - das Präsentation-Objekt der zuvor aktiven Zelle

Part: Controller

Dies wird benötigt um zu untersuchen, ob sich die Zeile der selektierten Zelle geändert hat und somit eine andere Zeile selektiert ist. Ist dies nämlich der Fall sollen erst dann die Änderungen die in der zuvor selektierten Zeile gemacht wurden gespeichert werden. Es wird die Methode 'saveData' aufgerufen, welche von den Komponenten, welche diese abstrakte Komponente einbinden, überschrieben werden muss.

Target-Component	Name
appWidgetAbstractTable	model.define.own.data.activeCellsPresentationEntry

model.observe.own.data.selectedTableEntriesFromOutside

Name: selectedTableEntriesFromOutside

Parameter: Array mit Präsentation-Objekten der Tabelle, welche in der Tabelle als selektiert markiert werden sollen

Part: Controller

Sagt dem Tabellenwidget, dass die übergebene Präsentations-Objekte in der Tabelle selektiert werden sollen (call auf 'table:selectedItems' mit dem gegebene Array als Übergabeparameter)

Target-Component	Name
appWidgetAbstractTable	model.define.own.data.selectedTableEntriesFromOutside

model.observe.own.data.selectedTableEntriesFromTable

Name: selectedTableEntriesFromTable

Parameter: Array mit Präsentation-Objekten der Tabelle, welche von der Tabelle als selektiert markiert wurden

Part: Controller

Aus den Präsentation-Objekten werden die dahinter liegenden Entitäten ausgelesen und sollen in das Modelfel der konkreten Komponente gesetzt werden, in dem die Selektion gehalten wird. Dafür wird die Methode 'selectedData' aufgerufen, welche von jeder konkreten Komponente überschrieben werden muss

Target-Component	Name
appWidgetAbstractTable	model.define.own.data.selectedTableEntriesFromTable

app.widget.table

Diese Komponenten kapselt den Umgang mit der Slickgrid Tabelle. Daten, Spalten und Interaktionen mit der Tabelle können mittels der Komponente beeinflusst werden. Damit soll eine Einheitliche Verwendung aller Tabellen gewährleistet werden und eine manuelle Einbinung von Slickgrid wird, außer in dieser Komponente, nicht mehr notwendig

API

API Definitions
api.register.table:columns
api.register.table:options
api.register.table:data
api.register.table:groupItemMetadataProvider
api.register.table:render
api.register.table:resize
api.register.table:changeRowCount
api.register.table:changedRows
api.register.table:scrollToIndex
api.register.table:setOnBeforeEditCellCallback
api.register.table:setOnBeforeMoveRowsCallback
api.register.table:selectPluginOptions
api.register.table:multiSelectPluginOptions
api.register.table:treePluginOptions
api.register.table:selectedItems

EVENTS

Events Definitions

events.publish.toParent.table:toggleTreeCollapse
events.publish.toParent.table:cellClicked
events.publish.toParent.table:rowsMoved
events.publish.toParent.table:activeCellChanged
events.publish.toParent.table:selectedDataItemsChanged
events.publish.toParent.table:dataChanged
events.publish.toParent.table:newRowAdded
events.publish.toParent.table:sortData
events.publish.toParent.table:startDragging
events.publish.toParent.table:onMouseEnter
events.publish.toParent.table:onMouseLeave
events.publish.toParent.table:drop
events.publish.toParent.table:dropstart
events.publish.toParent.table:dropend

MODEL

Model Definitions
model.define.own.data.dataView
model.define.own.data.data
model.define.own.data.columns
model.define.own.data.options
model.define.own.data.selectedCell
model.define.own.data.treeColumnWidth
model.define.own.data.singleSelectOptions
model.define.own.data.multiSelectOptions
model.define.own.data.treeOptions
model.define.own.event.selectedRowsChanged
model.define.own.event.cellClicked
model.define.own.event.dataChanged
model.define.own.event.newRowAdded
model.define.own.event.sortColumns
model.define.own.event.dragStarted
model.define.own.event.onDrop
model.define.own.event.onDropstart
model.define.own.event.onDropend
model.define.own.event.onMouseEnter
model.define.own.event.onMouseLeave
model.define.own.event.activeCellChanged
model.define.own.event.rowsMoved
model.define.own.command.renderGrid
model.define.own.command.resizeGrid
model.define.own.command.changeRowCount
model.define.own.command.changedRows
model.define.own.command.scrollRowToTop
model.define.own.command.scrollRowIntoView
model.define.own.command.setSelectedItems
model.define.own.command.setOnBeforeEditCellCallback
model.define.own.command.setOnBeforeMoveRowsCallback
model.observe.own.data.columns
model.observe.own.data.options
model.observe.own.data.treeColumnWidth
model.observe.own.event.cellClicked
model.observe.own.event.rowsMoved
model.observe.own.event.activeCellChanged
model.observe.own.event.selectedRowsChanged
model.observe.own.event.dataChanged
model.observe.own.event.newRowAdded

model.observe.own.event.sortColumns
model.observe.own.event.dragStarted
model.observe.own.event.onMouseEnter
model.observe.own.event.onMouseLeave
model.observe.own.event.onDrop
model.observe.own.event.onDropstart
model.observe.own.event.onDropend
model.observe.own.command.renderGrid
model.observe.own.command.resizeGrid
model.observe.own.command.changeRowCount
model.observe.own.command.changedRows
model.observe.own.command.setSelectedItems
model.observe.own.command.scrollRowToTop
model.observe.own.command.scrollRowIntoView
model.observe.own.command.setOnBeforeEditCellCallback
model.observe.own.command.setOnBeforeMoveRowsCallback

UI

UI Definitions

ui.plug.default

Details

api.register.table:columns

Name: table:columns

Parameter: Ein Array von slickgrid columns. Eine column muss dabei dem Format beschrieben im table-model.js -> dynamics -> singleColumnFormat folgen

Erlaubt das initiale Setzen der Tabellenspalten

Target-Component	Name
appWidgetAbstractTable	api.call.table:columns

api.register.table:options

Name: table:options

Parameter: Ein Object das dem beschriebenen Format im table-model.js -> dynamics -> optionsFormat folgt

Erlaubt das initiale Setzen der Tabellenoptionen

Target-Component	Name
appWidgetAbstractTable	api.call.table:options

api.register.table:data

Name: table:data
Parameter: Ein SlickGrid DataView Objekt oder Array
 Erlaubt das Setzen der DataView oder der Datenobjekten

Target-Component	Name
appWidgetAbstractTable	api.call.table:data

api.register.table:groupItemMetadataProvider

Name: table:groupItemMetadataProvider
Parameter: object
 Erlaubt das Setzen eines GroupItemMetadataProviders

Target-Component	Name
appWidgetAbstractTable	api.call.table:groupItemMetadataProvider

api.register.table:render

Name: table:render
 Schickt der Slickgrid Tabelle die Anforderung zum Neuzeichnen

Target-Component	Name
appWidgetAbstractTable	api.call.table:render

api.register.table:resize

Name: table:resize
 Schickt der Slickgrid Tabelle die Anforderung zum Neuberechnen der Höhe, Breite und Spaltenbreiten

Target-Component	Name
appWidgetAbstractTable	api.call.table:resize

api.register.table:changeRowCount

Name: table:changeRowCount
 Schickt der Slickgrid Tabelle die Anforderung den RowCount anzupassen und sich neu zu rendern

Target-Component	Name
appWidgetAbstractTable	api.call.table:changeRowCount

api.register.table:changedRows

Name: table:changedRows

Parameter: object

Schickt der Slickgrid Tabelle die Anforderung sich zu invalidieren, neu zu rendern und die Selektion zu synchronisieren.

Target-Component	Name
appWidgetAbstractTable	api.call.table:changedRows

api.register.table:scrollToIndex

Name: table:scrollToIndex

Parameter: number - Index der Reihe

Falls index 0 ist soll ganz nach oben gescrollt werden, ansonsten soll die Reihe in den Sichtbereich gescrollt werden.

Target-Component	Name
appWidgetAbstractTable	api.call.table:scrollToIndex

api.register.table:setOnBeforeEditCellCallback

Name: table:setOnBeforeEditCellCallback

Parameter: Eine Function, die der vorgegebenen Signatur von Slickgrid entspricht. Für Doku siehe: <https://github.com/mleibman/SlickGrid/wiki/Grid-Events>

Erlaubt das Eingreifen in den Editiermechanismus. Eine Editierung kann damit verhindert werden. Es muss dabei eine Callback-Function übergeben werden.

Target-Component	Name
appWidgetAbstractTable	api.call.table:setOnBeforeEditCellCallback

api.register.table:setOnBeforeMoveRowsCallback

Name: table:setOnBeforeMoveRowsCallback

Parameter: Eine Function, die der vorgegebenen Signatur von Slickgrid entspricht. Für Doku siehe: <https://github.com/mleibman/SlickGrid/wiki/Grid-Events> Achtung: onBeforeMoveRows ist ein Event auf dem Slickgrid Plugin 'RowMoveManager' und nicht auf Slickgrid

Erlaubt das Eingreifen in den Zeilenverschiebemechanismus des RowMoveManagers von Slickgrid. Es muss dabei eine Callback-Function übergeben werden.

Target-Component	Name
appWidgetAbstractTable	api.call.table:setOnBeforeMoveRowsCallback

api.register.table:singleSelectPluginOptions

Name: table:singleSelectPluginOptions
Parameter: Objekt
 Setzt die Optionen für das SingleSelect Plugin

Target-Component	Name
appWidgetAbstractTable	api.call.table:singleSelectPluginOptions

api.register.table:multiSelectPluginOptions

Name: table:multiSelectPluginOptions
Parameter: Objekt
 Setzt die Optionen für das MultiSelect Plugin

Target-Component	Name
appWidgetAbstractTable	api.call.table:multiSelectPluginOptions

api.register.table:treePluginOptions

Name: table:treePluginOptions
Parameter: Objekt
 Setzt die Optionen für das Tree Plugin

api.register.table:selectedItems

Name: table:selectedItems
Parameter: Objekt oder Array von Tabellen Items
 Schickt der Slickgrid Tabelle die Anforderung die gegebenen Items als selektiert zu markieren

Target-Component	Name
appWidgetAbstractTable	api.call.table:selectedItems

events.publish.toParent.table:toggleTreeCollapse

Name: table:toggleTreeCollapse
Parameter: Objekt mit den Attributen 'dataObject' gefüllt mit dem Datensatz der auf- oder
 zugeklappt werden soll 'columnId' Spaltenid, auf welcher der Toggle angestoßen wurde
 Wird ein Datenobjekt zusammen- oder aufgeklappt, wird dieser Event getriggert

events.publish.toParent.table:cellClicked

Name: table:cellClicked
Parameter: Objekt mit den Attributen 'dataObject' gefüllt mit dem Datensatz der auf- oder zugeklappt werden soll 'columnId' Spaltenid, auf welcher der Toggle angestoßen wurde
 Der Klick in eine Tabellenzelle wird mit diesem Event propagiert

events.publish.toParent.table:rowsMoved

Name: table:rowsMoved
Parameter: Objekt mit den Attributen 'items' gefüllt mit den Datensätzen die verschoben werden sollen 'insertBeforeItem' Der Datensatz vor welchem die items eingeschoben werden 'insertAfterItem' Der Datensatz nach welchem die items eingeschoben werden. Kann null sein.
 Sofern die Slickgrid Tabelle den RowMoveManager aktiviert, werden move events propagiert

events.publish.toParent.table:activeCellChanged

Name: table:activeCellChanged
Parameter: Objekt mit den Attributen 'dataObject' Datensatz, der aktuell den Fokus bekommt 'columnId' Spaltenid, welche den aktuellen Fokus bekommt 'row' Die neue aktive Zeile anhand ihrer zero-based Id 'cell' Die neue aktive Spalte anhand ihrer zero-based id
 Propagiert den Slickgrid event onActiveCellChanged

Target-Component	Name
appWidgetAbstractTable	events.subscribe.forChildren.table:activeCellChanged

events.publish.toParent.table:selectedDataItemsChanged

Name: table:selectedDataItemsChanged
Parameter: Array mit allen selektierten Datensätzen
 Liefert bei Veränderung der Selektion, die neu selektierten Datensätze

Target-Component	Name
appWidgetAbstractTable	events.subscribe.forChildren.table:selectedDataItemsChanged

events.publish.toParent.table:dataChanged

Name: table:dataChanged
Parameter: row Die Zeilennummer, in welcher verändert wurde cell Die Spaltennummer, in welcher verändert wurde
 Meldet die inhaltliche Veränderung einer Tabellenzelle durch den Nutzer

events.publish.toParent.table:newRowAdded

Name: table:newRowAdded
Parameter: item: Das Tabellen-Präsentations-Objekt columnId: Die Id der Spalte, in der eine Veränderung stattgefunden hat
 Meldet, dass eine Tabellenzeile hinzugefügt wurde

events.publish.toParent.table:sortData

Name: table:sortData
Parameter: Objekt mit den Attributen 'col': Die Spalte, anhand der sortiert werden soll 'asc': 1 oder -1 - die Richtung in die sortiert werden soll
 Meldet des Sort-Objekt

Target-Component	Name
appWidgetAbstractTable	events.subscribe.forChildren.table:sortData

events.publish.toParent.table:startDragging

Name: table:startDragging
Parameter: Objekt mit den Attributen 'data' Der Datensatz der verschoben werden soll restliche Slickgrid Event-Attributes werden zusätzlich durchgereicht
 Meldet den Datensatz, der bei Drag&Drop verschoben werden soll.

events.publish.toParent.table:onMouseEnter

Name: table:onMouseEnter
Parameter: Objekt mit den Attributen 'data' Datensatz, der aktuell unter dem Mauszeiger liegt 'columnId' SpaltenId, welche unter dem Mauszeiger liegt 'target' TODO: Klären! DOM Element welches als target für den MouseMove dient
 Propagiert den Slickgrid event onMouseEnter bei einer Zelle

events.publish.toParent.table:onMouseLeave

Name: table:onMouseLeave
 Propagiert den Slickgrid event onMouseEnter bei einer Zelle

events.publish.toParent.table:drop

Name: table:drop
Parameter: Objekt mit den Attributen 'dataObject' Datensatz, auf welchen gedroppt wird
'columnId' SpaltenId, auf welchen gedroppt wird 'droppedData' Die Daten, welche
gedroppt werden 'dropType' Die Art der Daten, welche gedroppt werden
Propagiert den Slickgrid event onDrop

events.publish.toParent.table:dropstart

Name: table:dropstart
Parameter: Objekt mit den Attributen 'dataObject' Datensatz, auf welchen gedroppt wird
'columnId' SpaltenId, auf welchen gedroppt wird 'domEv' TODO: Klären! Das
DOM Event des onDropStart 'dropType' Die Art der Daten, welche gedroppt werden
Propagiert den Slickgrid event onDropStart

events.publish.toParent.table:dropend

Name: table:dropend
Propagiert den Slickgrid event onDropEnd

model.define.own.data.dataView

Name: dataView
Parameter: Objekt
Hält ein SlickGrid DataView Objekt

model.define.own.data.data

Name: data
Parameter: Array
Hält die Tabellendaten

model.define.own.data.columns

Name: columns

Parameter: Array von Objekten mit der Signatur - siehe table-model.js -> dynamics -> singleColumnFormat

Menge der Slickgrid Spaltendefinitionen. Die Standard Attribute von SlickGrid einer Spalten wurden mit folgenden erweitert: - useFormatterForSearch: boolean (gibt an ob der Wert oder falls true der formatierte Wert für die Suche verwendet werden soll) - useFormatterForSort: boolean (gibt an ob der Wert oder falls true der formatierte Wert für das Sortieren verwendet werden soll) - validator: object (Eine Funktion zum Validieren des Spalten-Inhaltes) - previousWidth: number - tableDroppable: boolean (gibt an, ob in die Tabelle Elemente gedroppt werden können)

Target-Component	Name
widget-table	model.observe.own.data.columns

model.define.own.data.options

Name: options

Parameter: Objekt mit der Signatur - siehe table-model.js -> dynamics -> optionsFormat.

Menge der Slickgrid Tabellen Optionen. Die Standard Optionen von SlickGrid wurden mit folgenden erweitert: - dynRowHeight: boolean (gibt an ob sich die Höhe einer Reihe dynamisch am Inhalt anpassen soll) - activateRowSelectionModel: boolean (aktiviert die Selektion in der Tabelle allgemein. Wenn dieser Wert auf false gesetzt wird, haben die Optionen activateSelectPlugin und selectActiveRow keine Auswirkung) - activateSelectPlugin: boolean (aktiviert das Selekt-Plugin der Tabelle. Dadurch wird eine eigene Spalte Single oder Multi (abh. von der Option multiselect) in die Tabelle gerendert) - selectActiveRow: boolean (default: true - Ist der Wert false, die Selektion einer Tabellenzeile kann nur über die Spalte für die Selektion erfolgen und nicht über andere Spalten. Deswegen sollte selectActiveRow auf true gesetzt werden, wenn activateSelectPlugin auf false gesetzt wird.) - activateTreeTableFunctionality: boolean (es kann angegeben werden, wenn ein Baum dargestellt werden soll) - activateRowMoveManager: boolean (es kann angegeben werden, wenn es möglich sein soll, die Tabellenzeilen per Drag & Drop verschieben zu können.) - activateGrouping: boolean (es kann angegeben werden, wenn es möglich sein soll Den Tabelleninhalt anhand der einzelnen Spalten zu gruppieren.)

Target-Component	Name
widget-table	model.observe.own.data.options

model.define.own.data.selectedCell

Name: selectedCell

Parameter: Objekt

Die aktuell selektierte Zelle der Tabelle

model.define.own.data.treeColumnWidth

Name: treeColumnWidth

Parameter: number für die Breite in Pixeln

Beinhaltet die Breite der Treetable Collapse/Expand Spalte. Diese wird dynamisch verbreitert.

Target-Component	Name
widget-table	model.observe.own.data.treeColumnWidth

model.define.own.data.singleSelectOptions

Name: singleSelectOptions
Parameter: Objekt
 Optionen inklusive dem Markup für Singel-Selekt

model.define.own.data.multiSelectOptions

Name: multiSelectOptions
Parameter: Objekt
 Optionen inklusive dem Markup für Multi-Selekt

model.define.own.data.treeOptions

Name: treeOptions
Parameter: Objekt
 Optionen (Todo: Markup wie?) für eine Baumdarstellung

model.define.own.event.selectedRowsChanged

Name: selectedRowsChanged
Parameter: Array mit den selektierten Datensätzen
 Trigger wenn Tabellenselektionen verändert werden

Target-Component	Name
widget-table	model.observe.own.event.selectedRowsChanged

model.define.own.event.cellClicked

Name: cellClicked
Parameter: Objekt mit den Attributen 'cell' und 'cellId'
 Trigger wenn eine Tabellenzelle geklickt wird

Target-Component	Name
widget-table	model.observe.own.event.cellClicked

model.define.own.event.dataChanged

Name: dataChanged

Parameter: Objekt mit den Attributen: 'row' Zeilennummer der veränderten Zelle 'cell' Spaltennummer der veränderten Zelle

Trigger wenn eine Tabellenzelle verändert wurde

Target-Component	Name
widget-table	model.observe.own.event.dataChanged

model.define.own.event.newRowAdded

Name: newRowAdded

Parameter: Objekt mit den Attributen: 'item': Tabellen-Präsentations-Objekt 'column': Spalten-Objekt

Triggert, wenn eine neue Zeile hinzugefügt wurde, das gibt es nur wenn die Option 'enableAddRow' auf true gesetzt wurde.

Target-Component	Name
widget-table	model.observe.own.event.newRowAdded

model.define.own.event.sortColumns

Name: sortColumns

Parameter: Daten des Slickgrid Tabelle onSort Callbacks werden durchgereicht

Trigger wenn die Daten neu sortiert wurden

Target-Component	Name
widget-table	model.observe.own.event.sortColumns

model.define.own.event.dragStarted

Name: dragStarted

Parameter: Daten des Slickgrid Tabelle onDragStart Callbacks werden durchgereicht und um 'row' ergänzt

Trigger wenn Drag&Drop gestartet wurde

Target-Component	Name
widget-table	model.observe.own.event.dragStarted

model.define.own.event.onDrop

Name: onDrop

Parameter: Objekt mit den Attributen 'columnId', 'cell' und 'drop' Dabei stellt 'drop' die Referenz auf das jQuery Eventobjekt dar

Trigger wenn Drag&Drop auf einer Zelle durchzuführen ist

Target-Component	Name
widget-table	model.observe.own.event.onDrop

model.define.own.event.onDropstart

Name: onDropstart

Parameter: Objekt mit den Attributen 'columnId', 'cell', 'ev' und 'drop' Dabei stellt 'ev' die Referenz auf das jQuery Eventobjekt dar TODO: Klären was 'drop' macht

Trigger wenn ein Drag&Drop Element über eine Zelle geführt wird

Target-Component	Name
widget-table	model.observe.own.event.onDropstart

model.define.own.event.onDropend

Name: onDropend

Parameter: Das jQuery Eventobjekt wird weitergereicht

Trigger wenn ein Drag&Drop Element von einer Zelle wieder entfernt wird

Target-Component	Name
widget-table	model.observe.own.event.onDropend

model.define.own.event.onMouseEnter

Name: onMouseEnter

Parameter: Objekt mit den Attributen 'cell', 'target' und 'columnId'

Trigger wenn der Mauszeiger über eine Zelle geführt wird

Target-Component	Name
widget-table	model.observe.own.event.onMouseEnter

model.define.own.event.onMouseLeave

Name: onMouseLeave

Parameter: Boolean true

Trigger wenn der Mauszeiger von einer Zelle entfernt wird

Target-Component	Name
widget-table	model.observe.own.event.onMouseLeave

model.define.own.event.activeCellChanged

Name: activeCellChanged
Parameter: Objekt mit den Attributen 'cell', 'row' und 'columnId'
 Trigger wenn sich die aktive Zelle der Slickgrid Tabelle ändert

Target-Component	Name
widget-table	model.observe.own.event.activeCellChanged

model.define.own.event.rowsMoved

Name: rowsMoved
Parameter: Objekt mit den Attributen 'rows' und 'insertBefore'
 Trigger wenn sich Datensätze innerhalb der Tabelle verschoben haben durch den RowMoveManager

Target-Component	Name
widget-table	model.observe.own.event.rowsMoved

model.define.own.command.renderGrid

Name: renderGrid
 Schickt der Slickgrid Tabelle die Anforderung zum Neuzeichnen

Target-Component	Name
widget-table	model.observe.own.command.renderGrid

model.define.own.command.resizeGrid

Name: resizeGrid
 Schickt der Slickgrid Tabelle die Anforderung zum Neuberechnen der Höhe, Breite und Spaltenbreiten

Target-Component	Name
widget-table	model.observe.own.command.resizeGrid

model.define.own.command.changeRowCount

Name: changeRowCount

Schickt der Slickgrid Tabelle die Anforderung den RowCount anzupassen und sich neu zu rendern

Target-Component	Name
widget-table	model.observe.own.command.changeRowCount

model.define.own.command.changedRows

Name: changedRows

Parameter: object

Schickt der Slickgrid Tabelle die Anforderung sich zu invalidieren, neu zu rendern und die Selektion zu synchronisieren.

Target-Component	Name
widget-table	model.observe.own.command.changedRows

model.define.own.command.scrollToTop

Name: scrollToTop

Parameter: number - Index der Reihe

Scrollt die gegebene Reihe ganz nach oben.

Target-Component	Name
widget-table	model.observe.own.command.scrollToTop

model.define.own.command.scrollToView

Name: scrollToView

Parameter: number - Index der Reihe

Scrollt die gegebene Reihe in den sichtbaren Bereich.

Target-Component	Name
widget-table	model.observe.own.command.scrollToView

model.define.own.command.setSelectedItems

Name: setSelectedItems

Parameter: Object oder Array der zu selektierenden Objekte

Selektiert die gegebenen Datensätze

Target-Component	Name
widget-table	model.observe.own.command.setSelectedItems

model.define.own.command.setOnBeforeEditCellCallback

Name: setOnBeforeEditCellCallback
Parameter: Eine Function, die der vorgegebenen Signatur von Slickgrid entspricht. Für Doku siehe: <https://github.com/mleibman/SlickGrid/wiki/Grid-Events>
 setzt die onBeforeEdit Callback Funktion von SlickGrid

Target-Component	Name
widget-table	model.observe.own.command.setOnBeforeEditCellCallback

model.define.own.command.setOnBeforeMoveRowsCallback

Name: setOnBeforeMoveRowsCallback
Parameter: Eine Function, die der vorgegebenen Signatur von Slickgrid entspricht. Für Doku siehe: <https://github.com/mleibman/SlickGrid/wiki/Grid-Events>
 Setzt die onBeforeMoveRows Callback Funktion von SlickGrids RowMoveManager Plugin

Target-Component	Name
widget-table	model.observe.own.command.setOnBeforeMoveRowsCallback

model.observe.own.data.columns

Name: columns
Part: View
 Setzt der Slickgrid Tabelle neue Spaltendefinitionen und zeichnet die Tabelle erneut

Target-Component	Name
widget-table	model.define.own.data.columns

model.observe.own.data.options

Name: options
Part: View
 Setzt der Slickgrid Tabelle neue Tabellenoptionen und zeichnet die Tabelle erneut

Target-Component	Name
widget-table	model.define.own.data.options

model.observe.own.data.treeColumnWidth

Name: treeColumnWidth

Part: View

Updated die Spaltendefinition der TreeTable Spalte und zeichnet die Tabelle erneut

Target-Component	Name
widget-table	model.define.own.data.treeColumnWidth

model.observe.own.event.cellClicked

Name: cellClicked

Part: Controller

Triggert den Event 'table:cellClicked' oder 'table:toggleTreeCollapse' wenn ein klick auf den TreeTable Expand/Collapse erfolgt ist

Target-Component	Name
widget-table	model.define.own.event.cellClicked

model.observe.own.event.rowsMoved

Name: rowsMoved

Part: Controller

Triggert den Event 'table:rowsMoved'

Target-Component	Name
widget-table	model.define.own.event.rowsMoved

model.observe.own.event.activeCellChanged

Name: activeCellChanged

Part: Controller

Triggert den Event 'table:activeCellChanged'

Target-Component	Name
widget-table	model.define.own.event.activeCellChanged

model.observe.own.event.selectedRowsChanged

Name: selectedRowsChanged

Part: Controller

Triggert den Event 'table:selectedDataItemsChanged'

Target-Component	Name
widget-table	model.define.own.event.selectedRowsChanged

model.observe.own.event.dataChanged

Name: dataChanged
Part: Controller
 Triggert den Event 'table:dataChanged'

Target-Component	Name
widget-table	model.define.own.event.dataChanged

model.observe.own.event.newRowAdded

Name: newRowAdded
Part: Controller
 Triggert den Event 'table:newRowAdded'

Target-Component	Name
widget-table	model.define.own.event.newRowAdded

model.observe.own.event.sortColumns

Name: sortColumns
Part: Controller
 Triggert den Event 'table:sortData'

Target-Component	Name
widget-table	model.define.own.event.sortColumns

model.observe.own.event.dragStarted

Name: dragStarted
Part: Controller
 Triggert den Event 'table:startDragging'

Target-Component	Name
widget-table	model.define.own.event.dragStarted

model.observe.own.event.onMouseEnter

Name: onMouseEnter
Part: Controller
 Triggert den Event 'table:onMouseEnter'

Target-Component	Name
widget-table	model.define.own.event.onMouseEnter

model.observe.own.event.onMouseLeave

Name: onMouseLeave
Part: Controller
 Triggert den Event 'table:onMouseLeave'

Target-Component	Name
widget-table	model.define.own.event.onMouseLeave

model.observe.own.event.onDrop

Name: onDrop
Part: Controller
 Triggert den Event 'table:drop'

Target-Component	Name
widget-table	model.define.own.event.onDrop

model.observe.own.event.onDropstart

Name: onDropstart
Part: Controller
 Triggert den Event 'table:dropstart'

Target-Component	Name
widget-table	model.define.own.event.onDropstart

model.observe.own.event.onDropend

Name: onDropend
Part: Controller
 Triggert den Event 'table:dropend'

Target-Component	Name
widget-table	model.define.own.event.onDropend

model.observe.own.command.renderGrid

Name: renderGrid
Part: View
 Zeichnet die Tabelle erneut

Target-Component	Name
widget-table	model.define.own.command.renderGrid

model.observe.own.command.resizeGrid

Name: resizeGrid
Part: View
 Berechnet den Tabellen Viewport erneut und zeichnet die Tabelle erneut

Target-Component	Name
widget-table	model.define.own.command.resizeGrid

model.observe.own.command.changeRowCount

Name: changeRowCount
Part: View
 Berechnet den RowCount des Grids neu und rendert das Grid

Target-Component	Name
widget-table	model.define.own.command.changeRowCount

model.observe.own.command.changedRows

Name: changedRows
Parameter: object
Part: View
 Invalidiert das Grid, rendert das Grid und synchronisiert die Selektion

Target-Component	Name
widget-table	model.define.own.command.changedRows

model.observe.own.command.setSelectedItems

Name: setSelectedItems

Part: View

Markiert die gewünschten Datensätze in Slickgrid als 'selektiert'

Target-Component	Name
widget-table	model.define.own.command.setSelectedItems

model.observe.own.command.scrollToTop

Name: scrollToTop

Part: View

Scrollt die gewünschte Zeilennummer ganz nach oben

Target-Component	Name
widget-table	model.define.own.command.scrollToTop

model.observe.own.command.scrollToInView

Name: scrollToInView

Part: View

Scrollt die gewünschte Zeilennummer in den Sichtbereich der Tabelle

Target-Component	Name
widget-table	model.define.own.command.scrollToInView

model.observe.own.command.setOnBeforeEditCellCallback

Name: setOnBeforeEditCellCallback

Part: View

Setzt den Callback für den onBeforeEditCell Event von Slickgrid

Target-Component	Name
widget-table	model.define.own.command.setOnBeforeEditCellCallback

model.observe.own.command.setOnBeforeMoveRowsCallback

Name: setOnBeforeMoveRowsCallback

Part: View

Setzt den Callback für den onBeforeMoveRows Event von Slickgrids RowMoveManager Plugin

Target-Component	Name
widget-table	model.define.own.command.setOnBeforeMoveRowsCallback

ui.plugin.default

Name: default

Diese Komponente nutzt den Standard-Plug und kennt damit seine Umgebung nicht