# Grid

## Grid

Creates a new Grid, this contains columns, rows, and drag elements. Columns are vertical and can have n rows inside them. Rows are containers for divs, as the rows are resided, the div content will also be resided to fit.

## Constructor

```
new Grid(width, height, columns, rows)
```

Source: grid/grid.js, line 35

The Grid class takes in the window width and height, passed in to establish a basic window size. The default grid has no rows or columns within it. To add content call the Grid#createColumn method.

### Parameters:

| Name | Type | Description |
|---|---|---|
| `width` | `Number` | This is the width of the grid in pixels. |
| `height` | `Number` | This is the height of the grid in pixels. |
| `columns` | `Number` | This is the number of columns that the grid should be initialized with. |
| `rows` | `Number` | This is the number of rows that each column should have. |

## Methods

```
addColumn(column)
```

Source: grid/grid.js, line 362

This method can be called to add a column to the current grid. The grid is flexible, and has a constants size of 100%, when a new column is added, the default width of the column is (1/n)% where n is the number of columns in the grid. This method can be called at any time to add a new column.

### Parameters:

| Name | Type | Description |
| --- | --- | --- |
| column | Column | The Column that will be added. |

## addDrag(drag)

Source: grid/grid.js, line 392

This method lets the user register a new drag to appear on the window.

### Parameters:

| Name | Type | Description |
| --- | --- | --- |
| drag | H_Drag | The Horizontal drag to add to the window. |

## addWidget(widget)

Source: grid/grid.js, line 707

This function adds a widget into the grid of widgets being displayed

### Parameters:

| Name | Type | Description |
| --- | --- | --- |
| widget | Widget | This widget will be added into the grid at widget.col widget.row |

## createColumn(elements, properties) → {Column}

Source: grid/grid.js, line 133

Creates a Column. A column is a resizable container for data. Columns contain an index (left to right) starting at 0, representing which column they are numerically on the screen. Columns also contain a list of children

### Parameters:

| Name | Type | Description |
| --- | --- | --- |
| elements | Array | This is an array or a single html element. For every dom element passed, a new unique row will be created in this column |

| Name | Type | Description |
| --- | --- | --- |
| properties | Object | These are additional configuration parameters that can be passed into a column.<br>*Properties*<br>**Name** / **Type** / **Description**<br>color — String — A hexadecimal color to apply to the background of this column.<br>id — String — a string representing the ID that should be associated with this column |

### Returns:
A json object representing a Column.

Type Column

## createDrag(col1, col2) → {H_Drag}

Source: grid/grid.js, line 288

Creates a horizontal Drag. A horizontal drag is a small element conjoining two Columns. Clicking and dragging on a Drag will change the relative scale of the linked columns.

### Parameters:

| Name | Type | Description |
| --- | --- | --- |
| col1 | Column | The first column |
| col2 | Column | The second column |

### Returns:
Returns a json object representing a horizontal drag.

Type H_Drag

## createVDrag(row1, row2) → {V_Drag}

Source: grid/grid.js, line 330

Creates a vertical Drag. A vertical drag is a small element conjoining two rows. Clicking and dragging on a vertical Drag will change the relative scale of the linked rows.

### Parameters:

| Name | Type | Description |
| --- | --- | --- |
| row1 | Element | |
| row2 | Element | |

### Returns:
Returns a json object representing a Column.

Type V_Drag

## executeCallbacks()

Source: grid/grid.js, line 468

This function executes all registered callback functions inside of this grid.

## generateSaveObject() → {Object}

Source: grid/grid.js, line 536

This method is called when the editor is saving the active configuration. All columns and rows report their widths and heights respectivly. A multi-dimensional array object is constructed. This array has all of the height data needed to recreacte the current editor spacing on the next editor load.

### Returns:
sizes - The widths and heights of the current window configuration.

Type Object

## getCOLUMNS() → {Array.<Column>}

Source: grid/grid.js, line 585

Get all columns in this grid.

### Returns:
COLUMNS - The columns that make up this grid.

Type Array.<Column>

## getHEIGHT() → {Integer}

Source: grid/grid.js, line 577

Get the grid height.

### Returns:

HEIGHT - The height of the window.

Type Integer

## getWidget(col, row) → {Widget|null}

Source: grid/grid.js, line 678

This function returns the widget stored in grid cell (col, row). If there is no widget in that cell null is returned.

### Parameters:

| Name | Type | Description |
|---|---|---|
| col | Integer | This the column or X of the grid that you are trying to access. |
| row | Integer | This the row or Y of the grid that you are trying to access. |

### Returns:

widget - This is the widget in grid cell (col, row) if there is no widget in that cell, null is returned.

Type Widget | null

## getWIDTH() → {Integer}

Source: grid/grid.js, line 569

Get the grid width.

### Returns:

WIDTH - The width of the window.

Type Integer

## init(widgets)

Source: grid/grid.js, line 667

Initializes the grid with Widgets. A Widget is a synchronously spawned promise. This function will iterate through the widgets array and initialize one widget after another. This way later widgets can have earlier widgets passed into them. Example [Document(requires:[]), Tab(requires:[Document])] In this example the Widget Document requires nothing and is the first widget to be initialized in the program. Tab is the second widget to be initialized, and it requires Document. When Tab's initialize method is called, the value of Document will be stable and usable. This chaining method can be propagated forwards to allow widgets to require previously initialized widgets.

See Widget for information on built in widgets and how to create your own.

### Parameters:

| Name | Type | Description |
| --- | --- | --- |
| widgets | Array | This is an array of Widget elements that will be initialized in array order. |

## (async) initalize(widgets, COLUMNS, saveData) → {Promise.<void>}

Source: grid/grid.js, line 614

Asynchronous loop which will Synchronous populates a cell with an intialized widget untill all widgets are initialized, at this point it will then return.

### Parameters:

| Name | Type | Description |
| --- | --- | --- |
| widgets | Array.<Widget> | Uninitialized widgets to be initialized and loaded into the grid. |
| COLUMNS | Array.<Column> | The Columns that make up this EGAD grid. |
| saveData | Object | A save object representing the state of all widgets the last time the application was closed. |

### Returns:

- This promise resolves once all widgets are initialized.

Type Promise.<void>

## initializeWidgit(widget, saveData) → {Promise.<any>}

Source: grid/grid.js, line 596

This call wraps the widget promise constructor inside of another promise. The return value of this function is awaited upon inside of the initialize method.

### Parameters:

| Name | Type | Description |
| --- | --- | --- |
| widget | Widget | An uninitialized widget to initialize. |
| saveData | Object | An object containing information about the state of widgets the last time the application was closed.s |

### Returns:

This returns a promise wrapped around the widget's init function. The promise resolves when the widget finishes initializing.

Type Promise.<any>

## loadGridSizes(sizes)

Source: grid/grid.js, line 505

This method is called on editor load. It takes in an array of size configuration data. This data is used to position all grid elements such that they retain the positions they were in the last time the editor was closed.

### Parameters:

| Name | Type | Description |
|---|---|---|
| sizes | Array | The sizes of the grid elements. |

## onDrag(event, index1, index2)

Source: grid/grid.js, line 421

This function is called to whenever a horizontal drag event is triggered.

### Parameters:

| Name | Type | Description |
|---|---|---|
| event | Event | This is the drag event that was detected. |
| index1 | Integer | This is the index of the first column to be offset by this drag event. |
| index2 | Integer | This is the index of the second column to be offset by this drag event. |

## onDragEnd()

Source: grid/grid.js, line 481

This event is triggered after a horizontal drag has finished moving, it is used to set the absolute position of the 2 columns referenced by the drag event.

## onVDrag(event, row1, row2)

Source: grid/grid.js, line 443

This function is called to whenever a vertical drag event is triggered.

### Parameters:

| Name | Type | Description |
|---|---|---|

| Name | Type | Description |
| --- | --- | --- |
| event | Event | This is the drag event that was detected. |
| row1 | Integer | This is the index of the first row to be offset by this drag event. |
| row2 | Integer | This is the index of the second row to be offset by this drag event. |

## refresh()

Source: grid/grid.js, line 404

This function is called to reposition all of the drags and grid elements to proper positions after a drag has occurred, or any externalmovementt actions.

## removeWidget(widget)

Source: grid/grid.js, line 715

This function adds a widget into the grid of widgets being displayed

### Parameters:

| Name | Type | Description |
| --- | --- | --- |
| widget | Widget | This widget will be added into the grid at widget.col widget.row |

## resize()

Source: grid/grid.js, line 496

This function is used to sync the values of WIDTH and HEIGHT after the document window has been resided.

## setWidget(col, row, widget)

Source: grid/grid.js, line 693

This function changes the contents of cell (col, row) to widget. Null can be passed in to remove a widget from the grid, the row that that widget was in will persist however

### Parameters:

| Name | Type | Description |
| --- | --- | --- |
| col | Integer | This the column or X of the grid that you are trying to access. |

| Name | Type | Description |
| --- | --- | --- |
| row | Integer | This the row or Y of the grid that you are trying to access. |
| widget | Widget | This is the widget that cell (col, row) should be set to. |

*Documentation generated by JSDoc 3.5.5 on Sat Apr 13 2019 14:08:06 GMT-0400 (Eastern Daylight Time) using the docdash theme.*