



adges UI components library

**Component ID #1 – ex-media-grid - v1.0.0**

## Document Control

## Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Reason for Change |
| 1.0.0 | 10/20/2019 | Wael Gomaa |  |
| 1.1.0 | 09/24/2020 | Rijin Mukundan |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Review History

|  |  |  |
| --- | --- | --- |
| Version | Reviewer | Date |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table of Contents

[Document Control 1](#_Toc21195914)

[Version History 1](#_Toc21195915)

[Review History 1](#_Toc21195916)

[1. Document and components Purpose. 3](#_Toc21195917)

[2. General Information. 3](#_Toc21195918)

[3. How to use. 3](#_Toc21195919)

[4. Component features and configure. 4](#_Toc21195920)

[5. How to configure component. 4](#_Toc21195921)

[6. Examples 4](#_Toc21195922)

# 

# Document and components Purpose.

The purpose of the document is to guide how use and configure **ex-media-grid** component.

# General Information.

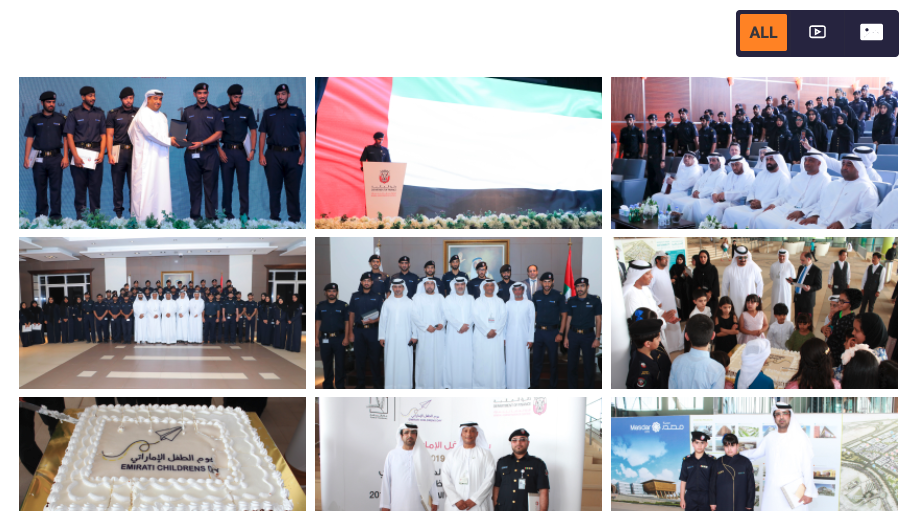
Component Name: Media-grid.

Extension Theme: ex-media-grid.

Owner: Wael Gomaa.

Version no: v1.0.0

Design



A screenshot of a cell phone

Description automatically generated

# How to use.

1. The available commands for this project .
   1. Compile : npm run gulp compile -- –path ex-media-grid –extension
   2. Compile & include under ADGE tenant : npm run gulp compile -- –path ex-media-grid –extension –full –dest [ADGEFolder]
   3. Deploy : npm run gulp deploy -- –path ex-media-grid –extension
2. *Now project have full copy from this extension theme under the project theme and you can apply the project brand guideline and configure component based on project needs.*

# Component features and configure.

You have access now to configure component with the below item.

$mediagrid-filters-wrapper-border : 1px solid $grey-900 !default;

$mediagrid-filters-button-container-bg:$grey-900 !default;

$mediagrid-filters-button-border-color: color-opacity($black-color , 0.19) !default;

$mediagrid-filters-button-bg-hover:$primary-color !default;

$mediagrid-filters-button-color:$white-color !default;

$mediagrid-filters-button-color-hover:$white-color !default;

$mediagrid-filters-button-bg-active:$primary-color !default;

$mediagrid-filters-button-color-active:$white-color !default;

$mediagrid-grid-item-background:$white-color !default;

$mediagrid-icon-color:$white-color !default;

$mediagrid-icon-color-hover:$white-color !default;

$mediagrid-grid-loadmore-mb: 50px!default;

$mediagrid-grid-gutter: 1% !default;

$mediagrid-grid-item-overlay-bg : $black-color !default;

$mediagrid-grid-item-overlay-opacity : 0.3 !default;

# How to configure component.

There are various ways to configure the component:

1. **CSS Class / Data Configuration (on the variant)**

**CSS Class Configuration Method**

These styles can be applied in the variant to get the desired result

|  |  |  |
| --- | --- | --- |
| Feature | Modifier | Default value |
| Without filter | ex-media-grid--without-filter |  |
| Without header | ex-media-grid--without-header |  |

**Data Attribute Modifiers**

These attributes can be applied in the variant to get the desired result

|  |  |
| --- | --- |
| Feature | Modifier |
| Enable gutter on the media grid , add this modifiers to .ex-media-grid\_\_body container | .ex-media-grid\_\_body--w-gutter |
| Media grid without filters control , you can use this modifier to the wrapper .ex-media-grid | .ex-media-grid--without-filter |
| Media grid without any header including the filters component , you can use this modifier to the wrapper .ex-media-grid | .ex-media-grid--without-header |
| To Enable filters for any DOM element just add the data-filter annotation | data-filter=’\*’ asterisk means all  data-filter=".classname " |
| To Enable media grid to any container use the data attribute data-media-grid | data-media-grid OR data-media-grid=”true” |
| To apply the grid to specific elements inside the grid use data-selector to avoid converting all childs to grid items | data-selector=itemCssClass' OR data-selector='. itemCssClass ' |
| To apply gutter between grid items add data-gutter attribute which value will refer to the gutter element | data-gutter='.gutterCssClass'  **Only apply if --w-gutter modifiers is applied** |
| To style the active filter class or change it add this data-active-filter-class with the your css class for the active status | data-active-filter-class="filterbuttonCssActiveClass"  **Default value is** **isotope-active** |
| To Enable fancybox to the items add data-fancybox-enabled | data-fancybox-enabled OR data-fancybox-enabled = “true” |
| To change the fancybox thumbs position add data-fancybox-thumbs-position | data-fancybox-thumbs-position = “right|bottom|left” **Default value is right** |
| To separate fancybox popup counter to the filtered elements use data-fancybox-mode attribute | data-fancybox-mode= ”none||separate”  **default value is none** |
| To show fancybox thumbs by default use data-fancybox-thumbs-autostart | data-fancybox-thumbs-autostart OR data-fancybox-thumbs-autostart=”true”  **default value is false** |
| To initialize grid with active filter use data-active-filter | data-active-filter=".videos" |

1. **Adding Built-In or Bootstrap Class (on the variant)**
   1. Spaces, Font weights, Display, Position properties etc can be added by bootstrap classes
      1. i. Spacing: <https://getbootstrap.com/docs/4.0/utilities/spacing/>
      2. ii. Flex: <https://getbootstrap.com/docs/4.4/utilities/flex/>
      3. iii. Font: <https://getbootstrap.com/docs/4.0/content/typography/>
   2. These classes should only be added to match the design
   3. For built-in classes check 7 Below to see the entire list
2. **SCSS Variable Override (on the \_override.scss file)**

To configure the component we can take the above variables from section **4. Component features and configure** and add them to: adge-ph2/themes/<ADGE>/vars/\_override.scss, this file with your desired changes to match the design.

1. **Complete style override**
   1. This should be the last and final option if the above 3 didn’t get you the desired output
   2. This is overwriting the SCSS of the component by adding your own style to it
   3. **Note: Take permission before doing this override**
   4. Instructions:
      1. Open the file /adge-ph2/themes/<ADGE>/sass/overrides.scss and add your own SCSS styles to override the faulty component.

# Dependencies

components-media-grid.css

jquery-fancybox-min.js

isotope-pkgd-min.js

lazysizes-min.js

fancybox-init.js

isotope-init.js

components-layout.js

components-media-grid.js

# Examples

Examples1:

I need to change text color.

I’ll copy the below variables from extension theme.

$highlight-main-tittle-color: $primary-color;

Then will paste the above variable in project folder theme in file \_override.scss and change with any variable based on site needs.

$highlight-main-tittle-color: $secondary-color;

Examples2:

I need to the button shape from square to had a rounded corner.  
I’ll copy the below variables from extension theme.

$button-shape-square: $border-radius-value;

$border-radius-value: 0px;

Then will paste the a above variable in project folder theme in file \_override.scss and change with any variable based on site needs.

$button-shape-square: $border-radius-value;

$border-radius-value: 20px;