



adges UI components library

**Component ID #1 – ex-master-card - v1.0.0**

## Document Control

## Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Reason for Change |
| 1.0.0 | 8/12/2020 |  |  |
| 1.1.0 | 24/09/2020 | 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-master-card** component.

The purpose of the dynamic highlight you can add any features or latest item on your site with image background or video and you can configure the component based on your branding guideline.

# General Information.

Component Name: Master-card.

Extension Theme: ex-master-card.

Owner: .

Version no: v1.0.0

Design

# How to use.

1. The available commands for this project .
   1. Compile : npm run gulp compile -- –path ex-master-card –extension
   2. Compile & include under ADGE tenant : npm run gulp compile -- –path ex-master-card –extension –full –dest [ADGEFolder]
   3. Deploy : npm run gulp deploy -- –path ex-master-card –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.

$master-card-border : 2px solid $grey-200 !default;

$master-card-icon-width: 35px !default;

$master-card-img-min-width:260px !default;

$master-card-img-max-width : 270px !default;

$master-card-img-md-max-width : 160px !default;

$master-card-img-md-min-width : 160px !default;

$master-card-img-icon-width:35px !default;

$master-card-type-icon-width:14px !default;

$master-card-img-margin:20px !default;

$master-card-details-margin-top: 20px !default;

$master-card-details-columns: 3 !default;

$master-card-details-md-columns: 2 !default;

$master-card-details-column-padding: 10px !default;

$master-card-hover-border-color: $primary-color !default;

$master-card-hover-border-bottom-width: 10px !default;

$master-card-hover-transform: translateY(-7px) !default;

$master-card-hover-transiton:0.5s !default;

$master-card-description-margin-top: 15px !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 |
| Master card horizontal | ex-master-card--horizontal |  |
| With hover effect on the card | ex-master-card--with-hover |  |

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.

# 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 \_custom.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 \_custom.scss and change with any variable based on site needs.

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

$border-radius-value: 20px;