



adges UI components library

**Component ID #1 – ex-happiness-meter - v1.0.2**

## Document Control

## Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Reason for Change |
| 1.0.0 | 11/10/2019 | Wael Gomaa |  |
| 1.0.2 | 29/12/2019 | Rana Abbassy | Adding variable |
| 1.0.2 | 23/09/2020 | 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-happiness-meter** component.

# General Information.

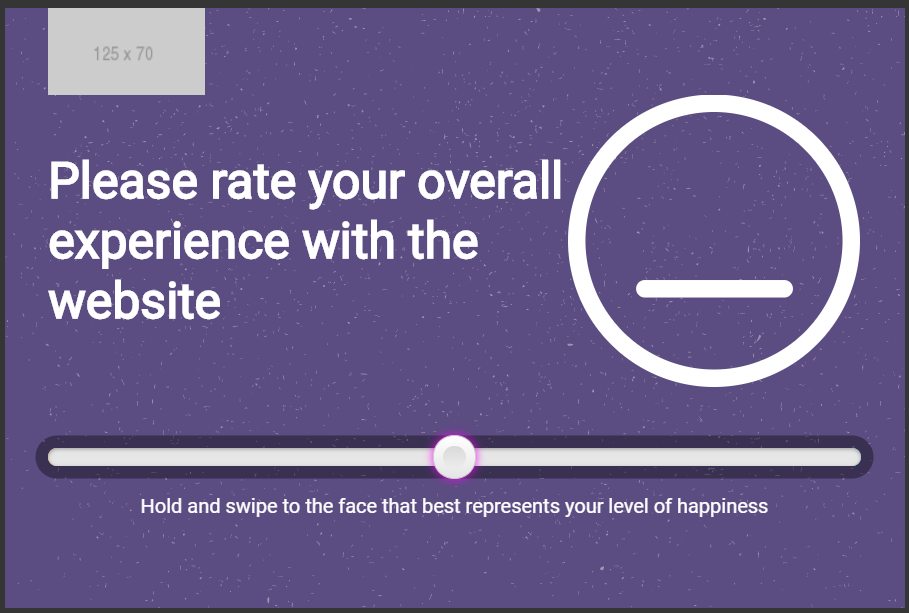
Component Name: Happiness-meter.

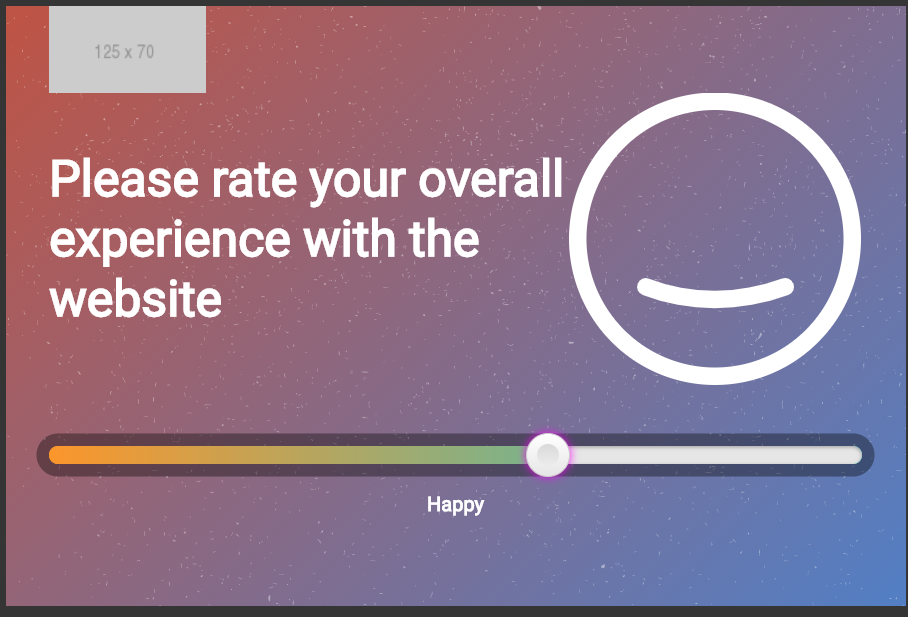
Extension Theme: ex-happiness-meter.

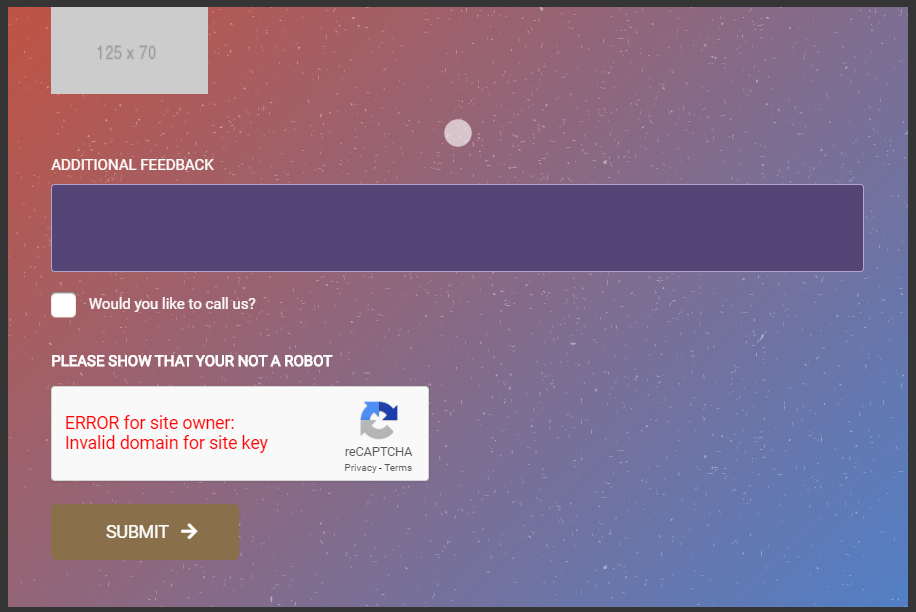
Owner: Rana Abbassy.

Version no: v1.0.2

Design







# How to use.

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

$happiness-meter-width : 720px !default;

$happiness-meter-height : 500px !default;

$happiness-meter-content-heading-color: $white-color !default;

$happiness-meter-content-heading-padding : 1rem !default;

$happiness-meter-padding : 0 !default;

$happiness-meter-firstbg : rgb(190,84,69) !default;

$happiness-meter-secondbg : rgba(82,127,197,1) !default;

$happiness-meter-initbg : #5B4C81 !default;

$happiness-meter-content-padding : 0 3rem 2rem 3rem !default;

$happiness-meter-form-padding : 15px 3rem !default;

$happiness-meter-header-border : 10px solid $white-color !default;

$happiness-meter-logo-padding : 20px !default;

$happiness-meter-logo-margin : 0 35px !default;

$happiness-meter-logo-max-width : 175px !default;

$happiness-meter-logo-background : $white-color !default;

$happiness-meter-logo-border-radius : 0px 0px 15px 15px !default;

$happiness-meter-face-size : 234px !default;

$happiness-meter-notice-color : $white-color !default;

$happiness-meter-notice-value : $white-color !default;

$happiness-meter-label-color : $white-color !default;

$happiness-meter-label-text-transform: uppercase !default;

$happiness-meter-icon-font-size : 55px !default;

$happiness-meter-logo-height : 55px !default;

$happiness-meter-error-color : #fdc8cc !default;

$happiness-meter-error-font-size: 12px !default;

$happiness-meter-input-bg: #534475 !default;

$happiness-meter-input-color: $white-color !default;

$happiness-meter-input-border : 1px solid #B1A5CC !default;

$happiness-meter-input-border-radius: 3px !default;

$happiness-meter-input-height : 40px !default;

$happiness-meter-input-padding: 15px !default;

$happiness-meter-textarea-height: 70px !default;

$happiness-meter-checkbox-label-color : $white-color !default;

$happiness-meter-checkbox-label-shadow-color1: $black-color;

$happiness-meter-checkbox-label-shadow-color2: $white-color;

$happiness-meter-checkbox-border : 3px solid #5B4C81 !default;

$happiness-meter-label-color : $white-color !default;

$happiness-meter-label-margin-bottom : 5px !default;

$happiness-meter-label-font-size :12px !default;

$happiness-meter-label-line-height : 20px !default;

$happiness-meter-slider-firstbg : #FC962B !default;

$happiness-meter-slider-secondbg : #26C6CA !default;

$happiness-meter-slider-horizontal-height: 15px !default;

$happiness-meter-slider-horizontal-width : 100% !default;

$happiness-meter-slider-horizontal-shadow-bg : color-opacity($white-color,0.1) !default;

$happiness-meter-slider-vertical-width : 15px !default;

$happiness-meter-slider-vertical-min-height : 150px !default;

$happiness-meter-slider-vertical-max-height : 100% !default;

$happiness-meter-slider-inactive-color: #e6e6e6 !default;

$happiness-meter-counter-color: $white-color !default;

$happiness-meter-counter-fontsize: 12px !default;

$happiness-meter-close-btn-color: color-opacity($white-color,0.6) !default;

$happiness-meter-close-btn-padding: 18px 20px !default;

$happiness-meter-close-btn-size: 16px !default;

$happiness-meter-slider-box-Shadow:$black-color !default;

$happiness-meter-slider-box-Shadow-fill:$black-color !default;

$happiness-meter-handle-bg-grdiant1:$white-color !default;

$happiness-meter-handle-bg-grdiant2:$black-color !default;

# How to configure component.

There are various ways to configure the component:

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

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

|  |  |
| --- | --- |
| Feature | Modifier |
| Applies Primary color styling | .ex-happiness-meter .happiness-meter-slider--horizontal |
| Applies Secondary color styling | .ex-happiness-meter .happiness-meter-slider--horizontal |

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-happiness-meter.css

components-happiness-meter-slider.css

components-floatingbutton.css

components.rangeslider.source.js

components-happiness-meter.js

components-rangeslider.js

Ex-main-them:

mooth-scrollbar-min.js

jquery-fancybox-min.js

jquery-validate-min.js

fancybox-init.js

scrollbar-init.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;