



adges UI components library

**Component ID History timeline - v1.0**

## Document Control

## Version History

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Reason for Change |
| **1.0** | March 2020 | Shaheen |  |
| **1.1** | September 2020 | Rijin Mukundan |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Review History

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

Table of Contents

Document Control 2

Version History 2

Review History 2

1. Document and components Purpose. 4
2. General Information. 4
3. How to use. 5
4. Component features and configure. 6
5. How to configure component. 7

# 

# Document and components Purpose.

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

# General Information.

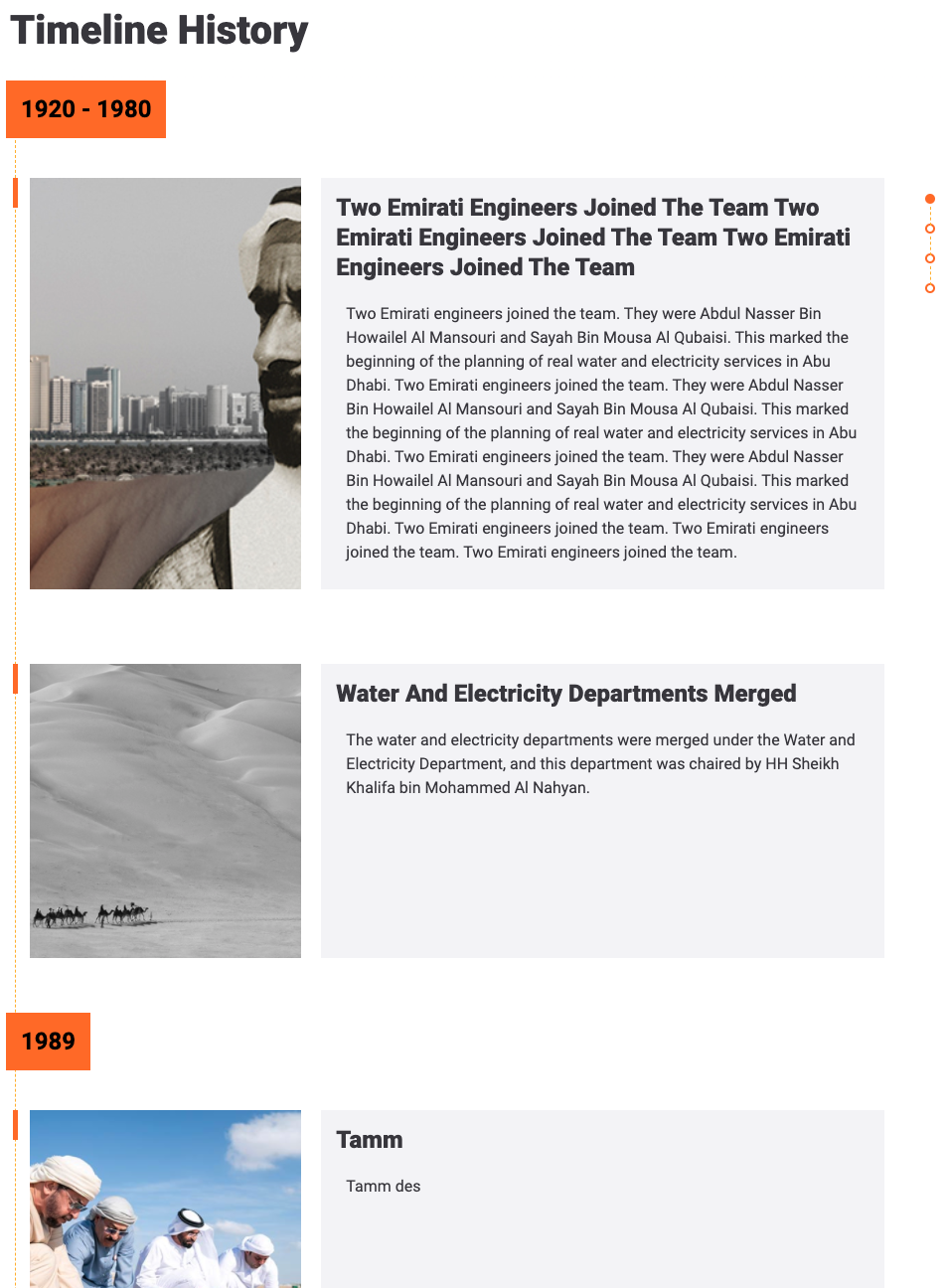
Component Name: Header

Extension Theme: ex-header.

Owner: Shaheen.

Version no: v1.0

Design



# How to use.

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

$history-timeline-border: 1px dashed $primary-color3 !default;

$history-timeline-journey-timeline-padding: 0 20px !default;

$history-timeline-primary-color: $primary-color2 !default;

$history-timeline-year-margin-top: 20px !default;

$history-timeline-year-margin-bottom: 20px !default;

$history-timeline-year-margin-left : -0.25rem !default;

$history-timeline-year-font-color: $black-color !default;

$history-timeline-year-hover-color: $black-color !default;

$history-timeline-year-padding: 14px 15px 14px 15px !default;

$history-timeline-content-padding: 10px !default;

$history-timeline-bullet-width: 5px !default;

$history-timeline-bullet-height: 30px !default;

$history-timeline-bullet-bg: $primary-color2 !default;

$history-timeline-bullet-alignment: 13px !default;

$history-timeline-content-bg: $grey-100 !default;

$history-timeline-mobile-image-min-height: 230px !default;

$history-timeline-nav-width: 100px !default;

$history-timeline-nav-top: 10% !default;

$history-timeline-nav-left: 35% !default;

$history-timeline-nav-bullet-bg: $white-color !default;

$history-timeline-nav-bullet-size: 3px !default;

$history-timeline-nav-bullet-radius: 50% !default;

$history-timeline-nav-bullet-spacing: 10px !default;

$history-timeline-nav-line-position-top: 8px !default;

$history-timeline-row-height : 351px !default;

*// readonly variables*

$history-timeline-row-padding: 20px;

# Dependencies

components-history-timeline.css

components-history-timeline.js

# How to configure component.

There are various ways to configure the component:

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.

Examples2:

I need to change the height of the header

$header-height: $header-height-value;

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

$header-height: 90px