**Component Ratingbar**

### **Document Properties**

| Item | Details |
| --- | --- |
| Document | Technical Specification Document |
| Application | Linear/ Donut Rating bar |
| Author | Pooja Shelar |
| Creation Date | 09/23/2020 |
| Last Updated | 09/23/2020 |
| Status | Completed |

**Contents**

[Document Properties](#_Toc51418511)

[**1. Component Overview**](#_Toc51418512)

[**1.1 Objective**](#_Toc51418513)

[**1.2 Use Cases**](#_Toc51418514)

[**1.3 Risks**](#_Toc51418515)

[**1.4 Out of Scope**](#_Toc51418516)

[**2. Technical Specifications**](#_Toc51418517)

[**2.1 NPM details**](#_Toc51418518)

[**2.2 NPM repo**](#_Toc51418519)

[**3.Unit Testing**](#_Toc51418520)

[**4. Assumptions**](#_Toc51418521)

**5.Concerns and Issue**s

1. Component Overview

* A rating bar is a component that is used to show ratings given by users to a particular thing.
* ORXe 3 has two distinct stylings for Rating Bar:
* Linear Progress Bar
* Donut Progress Bar

1.1 Objective

It consists mainly.

* As a standalone component, to allow the users to give ratings
* Rating bar displays an indicator showing the completion progress of a task, typically displayed as a progress bar.



**1.2 Use Cases**

There are two uses cases are present in rating bar component.

**Linear Progress Bar** - Linear Progress Bar shows the user rating in a linear manner.

Linear Progress Bar Component selector can be use as follow:

* Default rating bar -->

<orxe-rating-bar></orxe-rating-bar>

* Rating bar with linear type and a initial rating -->

<orxe-rating-bar type="linear" rating="60"></orxe-rating-bar>

* Rating bar with label -->

<orxe-rating-bar type="linear" rating="60" label="Experience 7.1 out of 10"></orxe-rating-bar>

Donut Progress Bar- Donut Progress Bar shows the user rating in a Circular manner.

Donut Progress Bar Component selector can be use as follow:

* Rating bar with donut type , initial rating and label -->

<orxe-rating-bar type="donut" rating="30" label="Experience 7.1 out of 10"></orxe-rating-bar>

* 1. Risks

NA.

1.4 Out of Scope

Defined certain styles for alignment and positioning for creating the base component.

2. Technical Specifications

1. The width of a linear progress bar depends on the available width of a mobile device. The text label and color of the linear progress bar carries a high contrast, in comparison with the background color, to pass WCAG 2.0 AA guidelines. The text below the linear progress bar shows the label and the rating in numbers.
2. The diameter of a donut progress bar is fixed. The text label and color of the donut progress bar carries a high contrast, in comparison with the background color, to pass WCAG 2.0 AA guidelines. The text inside the donut progress bar shows the rating in numbers.

### **Properties**

Properties which will be exposed by the component. Details of properties are as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name** | **Attribute** | **Access Modifier** | **Description** | **Type** | **Default** |
| Type | type | public | Type of rating bar . (linear/donut). | String | linear |
| Rating | rating | public | Actual value of the rating bar. | Number | 0 |
| Label | label | public | Used to provide a label for only linear rating bar. | String | '' |

* **Methods:**

Methods utilized by the component are as follows:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Signature** | **Access Modifier** | **Description** |
| getRating | \_getRating() : number {} | private | Converts the Rating of scale 1 - 100 by dividing it by 10. |
| getRotationClass | \_getRotationClass () : String{} | private | This method is used to get the appropriate rotation class |
| getLinearIndicatorColor | \_getLinearIndicatorColor ():TemplateResult {} | private | This method is used to return indicator color class based on ratings |
| addAriaLabel | \_addAriaLabel(): String{} | private | This method is used to add label attribute if it doesn’t exists by default it’s value is computed based on rating and label. |
| isOver50 | \_isOver50(): Boolean{} | private | This method is used to check whether rating is greater than 50 or not for Donut progress bar |
| \_linearRatingBar | \_inearRatingBar(): TemplateResult{} | private | This method is used to render Linear Rating bar. |
| \_ donutRatingBar | \_donutRatingBar (): TemplateResult{} | private | This method is used to render Donut Rating bar. |
| \_getIndicatorClass | \_getIndicatorClass(): String{} | private | This method is used to return class for linear rating bar based on ratings |
| render | render(): TemplateResult{} | public | This method is used to render the rating bar component |
| \_renderRatingBar | \_renderRatingBar(): TemplateResult{} | private | This method is used to render specific bar based on the type(linear/donut). |

### **Styling**

The following custom properties are available for styling:

|  |  |  |
| --- | --- | --- |
| **Custom Property** | **Description** | **Default** |
| --rating-bar-excellent | Used to set indicator color when ratingis above 8.5. | --primary |
| --rating-bar-great | Used to set indicator color when rating is in range 7- 8.5. | --rating-great |
| --rating-bar-average | Used to set indicator color when rating is in range 5.0-7.0. | --warning |
| --rating-bar-poor | Used to set indicator color when rating is in range 3.0-5.0. | --rating-poor |
| --rating-bar-bad | Used to set indicator color when rating is in range 1.0-3.0. | --error |

2.1 NPM details

Following packages are installed for project:

* “lit-element” – “^2.2.1”
* “lit-html” – “^1.1.2”
* “Orxe/cli version” - v2.0.20

Following command creating and running the component:

* Installing orxe/cli - npm install -g @orxe3/cli
* Create a new component workspace - orxe new component-workspace <name>
* Create a component – orxe g c ratingbar
* Serving the component – orxe serve

2.2 NPM Repository

1. Npm repository link –

2**.** npm package **–** npm install @ ratingbar / rating-bar @0.0.4

3. Used Render Selector - <orxe-rating-bar></orxe- rating-bar >

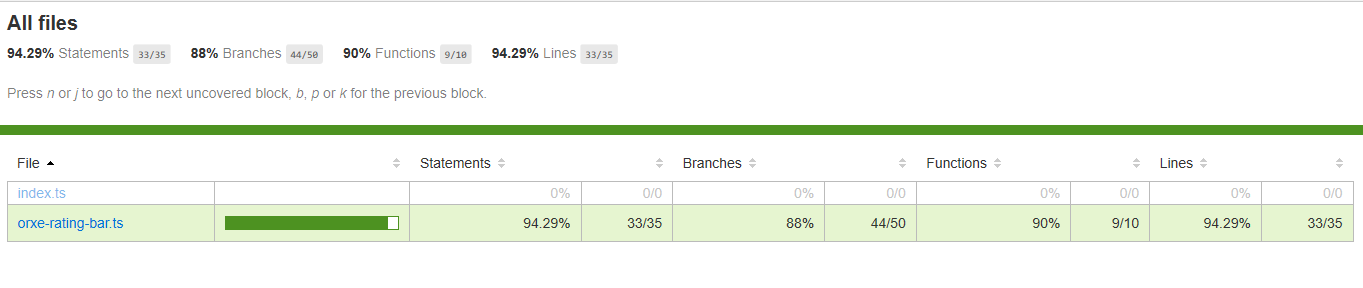
3.Unit Testing

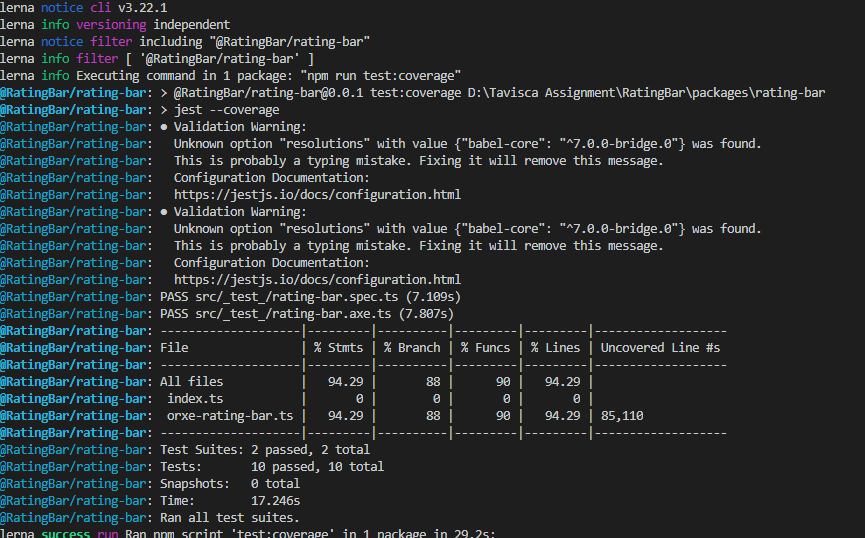
The test cases written for orxe- rating-bar component it consists two test suites. One is default rating-bar axe file and another rating-bar spec file. All function and line coverage are done with spec file.

All test cases of orxe- rating-bar component are written in rating-bar.axe.ts and rating-bar.spec.ts file. Test cases are available below.

1. Should create.
2. Should support all WCAG Accessibility Rules. When Default is rendered.
3. Should function render is call.
4. should function \_getRating returns rating in scale of 10'
5. should function \_getRotationClass returns rotation class
6. should function \_getLinearIndicatorColor returns rating color class
7. should function \_getIndicatorClass returns rating class
8. should check default value for properties for Rating bar
9. should function \_isOver50 returns true when rating is > 50
10. should function \_isOver50 returns false when rating is < 50

The unit test cases coverage report is as below.





4. Assumptions

This component is developed based on the design guidelines provided by Tavisca on the confluence page.

5. Concerns and Issues

NA.