**Component chips**

### Document Properties

| Item | Details |
| --- | --- |
| Document | Technical Specification Document |
| Application | chips |
| Author | Vikas Botre |
| Creation Date | 17/09/2020 |
| Last Updated | 21/7/2020 |
| Status | Completed |

Contents

1. Component Overview

**1.1 Objective**

**1.2 Use Cases**

**1.3 Risks**

**1.4 Out of Scope**

**2. Technical Specifications**

**2.1 NPM Details**

**2.2 NPM Repository**

**3. Unit Testing**

**4. Assumptions**

**5. Concerns and Issues**

# Component Overview

Chips are compact elements that represent an input, attribute, or action. Chips allow users to make selections, filter content, or trigger actions. The chips are currently placed directly on cards. They are placed horizontally in multiple numbers starting from the left-hand side with padding on the left. Chips are scrollable from right to left to see the chips which are hidden on the right side of the screen if there are any.

# Objective

The objective of the component, we can use the same component overall application. We have different web apps we can use the same guide, similar configurations, utilities.

# Use Cases

The application use cases are -

* **Selected** – Users can select single chips and process the data.

A picture containing text

Description automatically generated

* **Deselected** – Users can deselect the chips and process the data.

A picture containing text

Description automatically generated

# Risks

No risks

# Out of Scope

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

# Technical Specifications

If you plan to publish your package, the most important things in your package.json are the name and version fields as they will be required. The name and version together form an identifier that is assumed to be completely unique. Changes to the package should come along with changes to the version. If you don’t plan to publish your package, the name and version fields are optional.

Some rules:

* The name must be less than or equal to 214 characters. This includes the scope for scoped packages.
* The name can’t start with a dot or an underscore.
* New packages must not have uppercase letters in the name.
* The name ends up being part of a URL, an argument on the command line, and a folder name. Therefore, the name can’t contain any non-URL-safe characters.

# NPM Details

Following packages are installed for project:

“lit-element” – “^2.2.1”

“lit-html” – “^1.1.2”

Following command creating and running the component:

* Installing orxe/cli - npm install -g @orxe3/cli
* Create a new component workspace - orxe new component-workspace my-first-project
* Create a component – orxe generate c chips
* Serving the component – orxe serve

# NPM Repository

<https://packages.common.cnxloyalty.com/service/rest/repository/browse/npm-stage-> hosted/%40my-first-project/chips/

install npm packages: npm install @my-first-project/chips@0.0.5

User below selector in html: <orxe-chips></orxe-chips>

# Unit Testing

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

* Should check default attribute
* Should function render is call
* Should function addDynamicClass is call

The unit test cases coverage report is available below.

A screenshot of a video game

Description automatically generated

A screenshot of a computer

Description automatically generated

# Assumptions

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

# Concerns and Issues

No concern and issue.