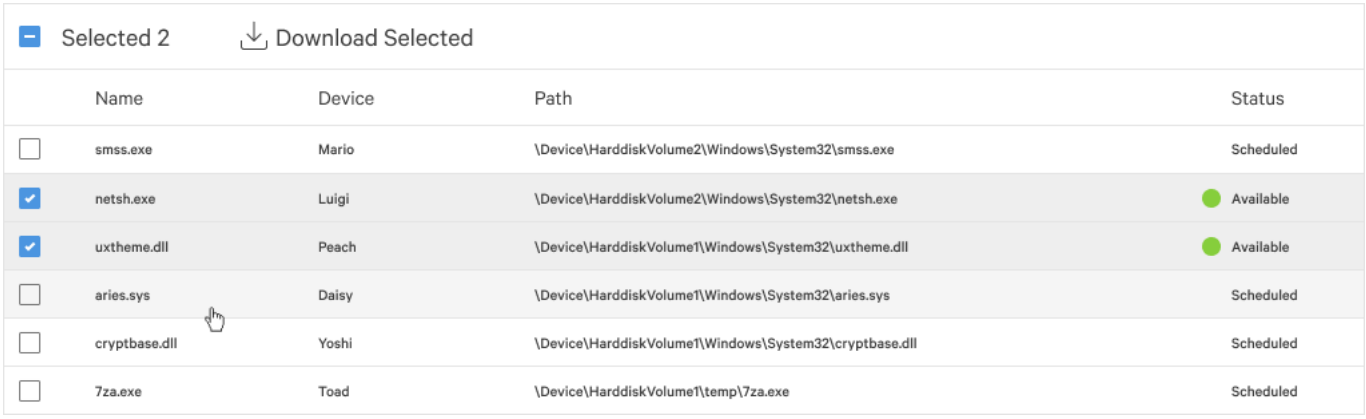
REUSABLE COMPONENT

COMPONENT TO BE IMPLEMENTED:



REQUIREMENTS:

- Only those that have a status of "available" are currently able to be downloaded. Your implementation should manage this.
- The select-all checkbox should be in an unselected state if no items are selected.
- The select-all checkbox should be in a selected state if all items are selected.
- The select-all checkbox should be in an indeterminate state if some but not all items are selected.
- The "Selected 2" text should reflect the count of selected items and display "None Selected" when there are none selected.
- Clicking the select-all checkbox should select all items if none or some are selected.
- Clicking the select-all checkbox should de-select all items if all are currently selected.
- Status should be correctly formatted
- Clicking "Download Selected" when some or all items are displayed should generate an alert box with the path and device of all selected files.
- Precise/exact HTML formatting/styling to match the mockup is not required however rows should change colour when selected and on hover.

IMPLEMENTATION:

The component was developed using the Angular 17 framework, without incorporating any external libraries or CSS frameworks. It leverages extensive JavaScript, TypeScript, and CSS features, with SCSS utilized for enhanced readability. Jest was employed to write the test cases for the assignment.

STEPS TO RUN THE PROJECT:

1. Ensure Node.js is installed on the local system.
2. Clone the repository using `git clone https://github.com/sugandhaduhan23/resusable-grid.git'.
3. Run `npm install` to install all required packages.
4. Use `ng serve --open` to run and automatically open the application in the browser. Alternatively, `npm run start` can also be used to start the application. The application will run on http://localhost:4200 by default.
5. Execute `npm run test` to run all the test cases.

IMPLEMENTATION

The component has been developed as a reusable entity with dynamic column names, fields, and configuration. The focus of the implementation is reusability with no hardcoded values, accessibility, scalability and testability. It can be incorporated into any other component using the selector defined in the `.ts` file. The component can be used in any other component using <ui-grid></ui-grid> custom tag.

FEATURES

Select All Functionality:

- Selecting "Select All" will check all checkboxes.
- Selecting some rows will set the "Select All" checkbox to an indeterminate state.
- If "Select All" is checked, unselecting it will deselect all rows.
- If the user manually selects all rows, the "Select All" checkbox will automatically be checked.

Selected Items Count:

- Display the count of selected rows in the top section.
- Display "None Selected" if no rows are selected.

Row Interaction:

- Rows change color when selected or hovered.

Download Button:

- The "Download Selected" button remains disabled if no rows are selected.

Dynamic Column Names:

- Column names can be passed dynamically with various configurations such as class, width, pipe, hide, headerClass, etc.

Configurations:

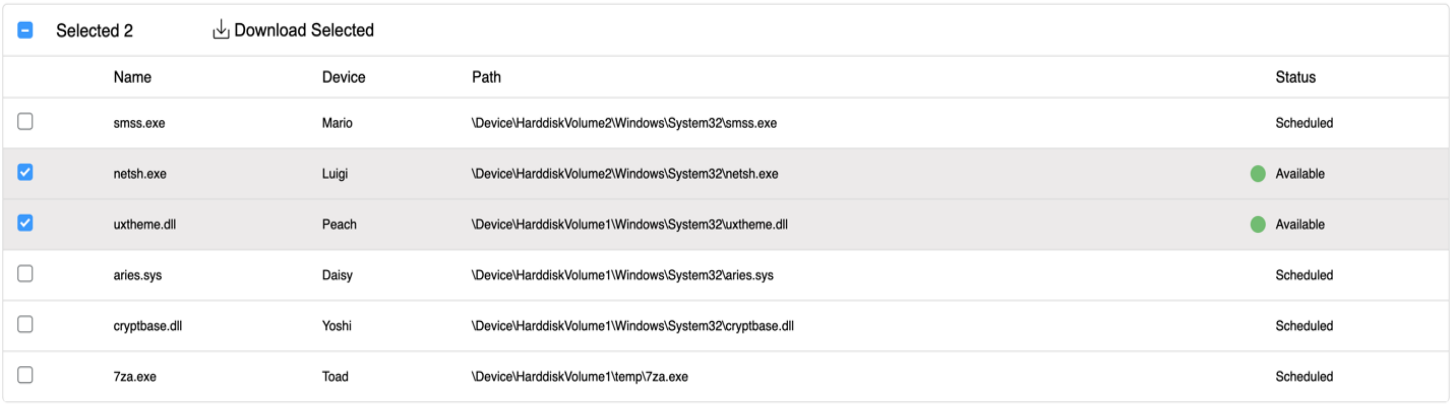
- Configurations like showHeader, hideDownload, headerCheckboxSelection, etc., can be passed to enable/disable or show/hide different features.

Checkbox Disabling:

- A property named “disableCheckbox” in the table configurations accepts a function to disable row checkboxes if necessary.

APPROACH 1

- A table with all checkboxes enabled, allowing users to select any data.
- Clicking the "Download Selected" button filters the data based on its status within the component, ensuring reusability.

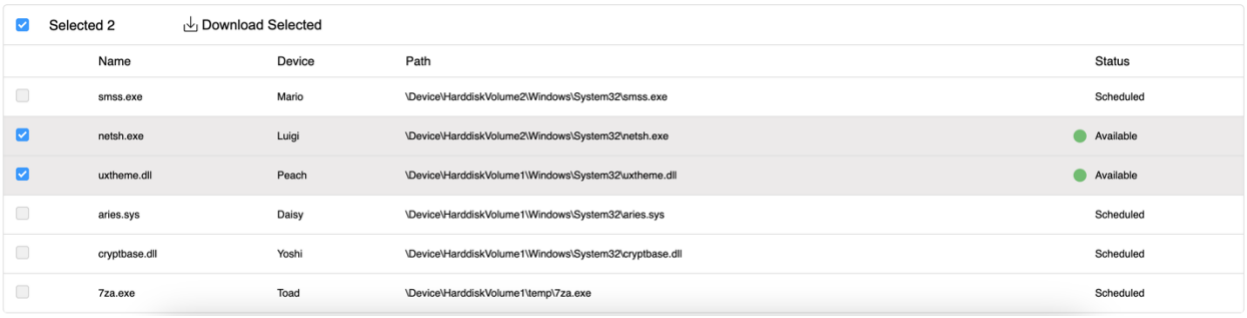


APPROACH 2

- A table that disables checkboxes if the status is anything other than 'Available'.

- If the table configuration includes the disableCheckbox option, a function is invoked within the parent component where you can define a condition to disable a row.

  Note: Disabled rows are excluded from the "Select All" functionality.



ALERT-BOX

A reusable alert box component has been created instead of using the native JavaScript alert box. It allows embedding dynamic content in the alert box via HTML in the parent component.

Note:

Two different components using the same table are created to showcase both implementations in the project:

1. device.component: Basic implementation.
2. device1.component: Implementation with disableCheckbox configuration.

VIDEO EXPLANATION LINK:
https://drive.google.com/file/d/1j35XNqNLElykd7DEo4seqQ5lwPeroBp0/view?usp=sharing