This is the documentation to show the usage of the custom made library **Dynamic-Grid.** This library is helpful to generate a dynamic grid with the dynamic columns and rows also with the custom configuration.

**Usage:**

The Grid component can be generated by passing below parameters to make it customize.

1. Data -> This should be a dynamic JSON Object (from API/local) which contains the headers and the grid body data in below format.

const data = {

headers: [{key:’headerKey’, name:’headerValue’}],

rowData:[{id:’1’, name:’neeraj’}]

}

1. maxItemToShowOnPage -> this property is to support the pagination on the dynamic-grid. Based on the data length and this property value the pages will be created.
2. transformHeaderValue ->This is a custom function to be used when there is any need to provide the custom transformed value for the header name.
3. transformCellData -> This is a custom function to be used when there is any need to transform the value of any cell data.
4. renderGridHeader -> Custom function to provide the default/customized html for the grid header cell.
5. renderCell-> Custom function to provide the customized html for the grid data cell.
6. cellOrderKey: This is an array of the header keys to rearrange the column order of the grid.

Sample example:

const GridComponent = new DynamicGrid({

data: [{ headers: [{key:'id',name:'ID'},{key:'age',name:'Age'}], rowData: [] }],

maxItemToShowOnPage: 10,

transformHeaderValue: ({headerKey, headerValue}) => { return `<h1>${headerValue}</h1>` },

transformCellData: (cellKey, cellValue) => { return `<h1>${cellValue}</h1>` },,

renderGridHeader: (headerKey, headerValue) => { return `<h1>${cellValue}</h1>` },,,

renderCell: (cellKey, cellValue) => { return `<h1>${cellValue}</h1>` }

cellOrderKeys: ['age', 'id']

});