# NG-TABLE LIBRARY

NG-TABLE LIBRARYis a library for adding tables in our project.

**Features Available:**

* Pagination
* Sorting
* Items Per Page
* Whole search and Column Search
* Go to particular Page
* Action column(View, Edit, Delete, Send button etc)

**How to use it?**

1. Create a New folder in your Angular project and name it as **projects**.
2. Add the **library folder (E.g.: ng-custom-table**folder**)** inside the projects folder.
3. Add the following code in your **angular.json** file inside the projects alongside of your application.

"ng-custom-table": {

      "root": "projects/ng-custom-table",

      "sourceRoot": "projects/ng-custom-table/src",

      "projectType": "library",

      "prefix": "ng",

      "architect": {

        "build": {

          "builder": "@angular-devkit/build-ng-packagr:build",

          "options": {

            "tsConfig": "projects/ng-custom-table/tsconfig.lib.json",

            "project": "projects/ng-custom-table/ng-package.json"

          }

        },

        "test": {

          "builder": "@angular-devkit/build-angular:karma",

          "options": {

            "main": "projects/ng-custom-table/src/test.ts",

            "tsConfig": "projects/ng-custom-table/tsconfig.spec.json",

            "karmaConfig": "projects/ng-custom-table/karma.conf.js"

          }

        },

        "lint": {

          "builder": "@angular-devkit/build-angular:tslint",

          "options": {

            "tsConfig": [

              "projects/ng-custom-table/tsconfig.lib.json",

              "projects/ng-custom-table/tsconfig.spec.json"

            ],

            "exclude": [

              "\*\*/node\_modules/\*\*"

            ]

          }

        }

      }

    }

1. Import the Library Module in the **Module** file.

import { NgCustomTableModule } from 'projects/ng-custom-table/src/lib/ng-custom-table.module';

Also addNgCustomTableModule to the imports array in the same file.

1. Add the following in the **tsconfig.json** file.

"resolveJsonModule": true,

"esModuleInterop": true

This allows us the json files import in the ts.

****

1. Install bootstrap, jquery, ngx-pagination, ngx-order-pipe, font-awesome icons and popper npm.

npm install - - save bootstrap

npm install - - save jquery

npm install - - save font-awesome angular-font-awesome

npm install - - save popper.js

npm install - - save ngx-pagintion

npm install - - save ngx-order-pipe

1. Add the following in your project array in the **angular.json** file.

* In **styles** Array,

"./node\_modules/bootstrap/dist/css/bootstrap.min.css",

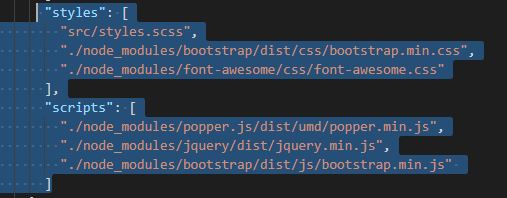
"./node\_modules/font-awesome/css/font-awesome.css"

* In **scripts** Array,

"./node\_modules/popper.js/dist/umd/popper.min.js",

"./node\_modules/jquery/dist/jquery.min.js",

"./node\_modules/bootstrap/dist/js/bootstrap.min.js"



1. Go to the component where you want to add the table.

**In html file:**

Add the below code

<ng-custom-table [tableconfig]="tableconfig" (actionOutput)="onActionClick($event)"

(searchOutput)="onSearchEvent($event)"

(onChangeInTable)="onchangeTable($event)"></ng-custom-table>

Here,

**tableconfig** is a interface that contains the following fields.

* actionSetting

**actionSetting** is an interface which defines the action column in the table. It contains the following fields:

* showActionColumn

This is a Boolean value that works in Show/Hide the action column.

* width

This is an optional field. It’s used to determine the width of the Actions Column

* include\_actions

This is an optional field. It’s used in determining what actions we do want to show in the Actions Column

* displayAll

**displayAll** is a Boolean value, which when true shows all the records in a table in the single page. It removes pagination, items per page and go to particular page when trued.

* columns

**columns** is an array of json which contains the following keys along with its value pair.

* title

This is the column name that is displayed in the table.

* sort

This is a Boolean value that determines if sort is needed for a particular column.

* columnObj

This is the value printed on the table. This is assigned with the keys from the json data.

* filterString

This is a Boolean value to enable/disable filter for individual columns.

* width

This is used to set width of the individual columns.

* inputType

This is used to pass html or component or string as a column.

* tableData

tableData holds the json structure of the data in the table.

* sortByColumn

This field is used to set the default sort on load based on a single column.

* sortingOrder

sortingOrder is used to set the initial sort of the table. It’s an optional Field. By default it is set to ascending.

* itemsPerPage

itemsPerPage is to set number of records to be displayed per page. This is an optional field and the default value is 10.

* title

title is used to name the table. It’s also used in pagination to set id.

* backendSearch

backendSearch is a Boolean value to set the table search based on frontend or backend.

* totalDataInBackend

totalDataInBackend specifies the total number of records in the backend. This is used when backendSearch is set to true.

**actionOutput**

actionOutput passes the action clicked along with the row and column data to the parent component.

**searchOutput**

searchOutput passes the searched value(the value entered in the search box) to the parent component.

**onChangeInTable**

onChangeInTable passes the values to the parent Component if there’s a change in data after pagination and sorting.

<ng-custom-table [tableconfig]="tableconfig" (actionOutput)="onActionClick($event)"

    (searchOutput)="onSearchEvent($event)" (onChangeInTable)="onChangeInTable($event)">

</ng-custom-table>

**In TS File:**

Include the imports and declare the variables that are needed to be passed to the child component.

Note: Try to auto-import the data rather than copy-pasting the import statements.

Add the below code in the ts file:

//import the interface from the library

import { Component, OnInit } from '@angular/core';

import { tableConfig } from 'projects/ng-custom-table/src/public-api';

importfruitjson from './data.json'; //set path for json if it is in any other folder

import { PassingcomponentComponent} from './passingcomponent/

passingcomponent.component'; //to pass component as a column

//variable declaration

tableconfig: tableConfig;

publicfruitjson: any = fruitjson;

//passing values (Add it in ngOnInit())

this.tableconfig = {

"actionSetting": {

"showActionColumn": true,

"width": '15%',

"include\_actions": ["Edit", "Delete", "Enable", "Disable", "View"]

},

"displayAll": false,

"columns": [

{ "title": "Fruit ID", "sort": true, "columnObj": "id", "filterString": true, "width": "15%" },

{ "title": "Name", "sort": true, "columnObj": "name", "filterString": true, "width": "15%" },

{ "title": "Description", "sort": true, "columnObj": "description", "filterString": true,

"width": "25%"},

{ "title": "Availability", "sort": false, "columnObj": "availability", "filterString": true,

"width": "15%" ,"inputType":"Component"},

{ "title": "Price", "sort": false, "columnObj": "cost", "filterString": false, "width": "15%" }

],

"tableData": this.fruitjson.fruits,

"sortBycolumn": "id",

"sortingorder": "Asc",

"itemsPerPage": "5",

"title": "FruitsTable",

"backendSearch": true,

"totalRecordsInBackend": 75

}

//Sample Function to get Table Values based on clicked action.

onActionClick(data) {

//Perform the required action based on data

}

//Receives the searched data after filtering

onSearchEvent(searchData) {

console.log(searchData);

}

//Receives data that is changed based on pagination and sorting.

onchangeTable(data) {

console.log(data)

}

//To pass a component in the column

this.tableconfig.tabledata.foreach(data =>

data.availability = PassingcomponentComponent

})

}

Here,

* tableConfig

It is the variable that passes the values in the tableConfig to the child Component. The available values of the fields in the interface can be viewed by hovering over the variable.

* actionClick()

This is a user defined function to get the action and the data that is clicked on the table.

* onSearchEvent()

This is a user defined function which is used to receive the search data based on the search.

* onchangeTable()

This is a user defined function which is used to receive the values that change based on changes like pagination and sorting.

import { Component, OnInit } from '@angular/core';

import { tableConfig } from 'projects/ng-custom-table/src/public-api';

import fruitjson from './data.json';

import { PassingcomponentComponent} from './passingcomponent/

passingcomponent.component';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrls: ['./app.component.scss']

})

export class AppComponent implements OnInit {

  tableconfig: tableConfig;

  public fruitjson: any = fruitjson;

  ngOnInit() {

    this.tableconfig = {

      "actionSetting": {

        "showActionColumn": true,

        "width": '15%',

        "include\_actions": ["Edit", "Delete", "Enable", "Disable", "View"]

      },

      "displayAll": false,

      "columns": [

        { "title": "Fruit ID", "sort": true, "columnObj": "id", "filterString": true, "width": "15%" },

        { "title": "Name", "sort": true, "columnObj": "name", "filterString": true, "width": "15%" },

        { "title": "Description", "sort": true, "columnObj": "description", "filterString": true,

        "width": "25%"},

        { "title": "Availability", "sort": false, "columnObj": "availability", "filterString": true,

        "width": "15%" ,"inputType":"Component"},

        { "title": "Price", "sort": false, "columnObj": "cost", "filterString": false, "width": "15%" }

      ],

      "tableData": this.fruitjson.fruits,

      "itemsPerPage":"5",

      "sortBycolumn":"id",

      "sortingorder":"Asc",

      "backendSearch": false,

      "totalRecordsInBackend": 75

    }

    this.tableconfig.tableData.forEach(data=>{

      data.availability = PassingcomponentComponent

    })

  }

  onActionClick(data) {

    console.log(data);

  }

  onSearchEvent(searchData) {

    console.log(searchData);

  }

  onchangeTable(Data) {

    //console.log(Data);

    })

  }

**In style.scss:**

Add the following in the style.scss file. The variables **$sorting-arrow-theme**, **$table-header-background-theme** and**$table-header-text-theme** are used to set colours for arrow in sorting, table header and table header text.

$sorting-arrow-theme:black;

$table-header-background-theme:white;

$table-header-text-theme:black;

**In the Library:**

**ng-custom-table.component.ts** is the file that contains the actions. So if there is anything to be changed (like adding actions, changing color or icons) we can do it in this file. (**Folder**: ng-custom-table)

**References:**

1. <https://github.com/valor-software/ng2-table>
2. <https://github.com/akveo/ng2-smart-table>
3. <https://angular.io/api/platform-browser/DomSanitizer>
4. <https://www.npmjs.com/package/ngx-pagination>
5. <https://www.positronx.io/angular-checkbox-tutorial/>
6. <https://www.typescriptlang.org/docs/handbook/interfaces.html>
7. <https://www.typescriptlang.org/docs/handbook/enums.html>
8. <https://devhints.io/jsdoc>