

Angular JS Custom Directive for Grid

(Reusable component for all the projects)

VERSION CONTROL

Prepared by:	Manoharan, Naresh
Date:	5/11/2015
Reviewed and Accepted by:	B.C.Subramanian
Date:	6/11/2015
Approved by:	Baskaran.Varadarajan, D.A.Soundararajan
Date:	29/12/2015

VERSION HISTORY

Date	Version	Author	Description
5/11/2015	0.1	Manoharan, Naresh	Initial Draft Version
9/11/2015	1.0	B.C.Subramanian	Final Reviewed
21/12/2015	1.1	S.Kalaiyarasu	Reviewed
28/12/2015	1.2	Baskaran.Varadarajan	Final Approved

Contents

1. INTRODUCTION TO ANGULAR JS CUSTOM DIRECTIVE FOR GRID	4
1 Introduction.....	4
1.1 Purpose	4
1.2 Supporting elements by angular js to create directive for Grid	4
1.3 Intended Audience	5
1.4 Defining a directive.....	5
1.5 Sample code	6
1.5.1 ANGULAR DIRECTIVE CODE	6
1.5.2 HTML VIEW CODE	9
2. GRID DIRECTIVE INFORMATION USING ANGULAR JS.....	10
2.1 View (Sample data)	10
2.2 general information about the directive	11
2.3 DB JSON object structure	12
2.4 JSON object variable details	13
2.5 Current issues and Addresses	15
2.6 Pros and Cons.....	15

1. INTRODUCTION TO ANGULAR JS CUSTOM DIRECTIVE FOR GRID

1 INTRODUCTION

Custom directives are used in AngularJS to extend the functionality of HTML. Custom directives are defined using "directive" function. A custom directive simply replaces the element for which it is activated.

AngularJS application during bootstrap finds the matching elements and do one time activity using its compile () method of the custom directive then process the element using link () method of the custom directive based on the scope of the directive.

Directives are markers on a DOM element (such as an attribute, element name, comment or CSS class) that tell AngularJS's HTML compiler (\$compile) to attach a specified behaviour to that DOM element (e.g. via event listeners), or even to transform the DOM element and its children.

- Grid directive is a data grid for AngularJS, customized version of Angular UI Grid.
- Includes features like Customizable templates, Sorting, Filtering, User interaction, Column Pinning, Grouping, Inline Editing, Expandable rows, Drag gable rows

1.1 PURPOSE

- The Grid directive is mostly used for large data sets, performs well even 10,000+ rows

1.2 SUPPORTING ELEMENTS BY ANGULAR JS TO CREATE DIRECTIVE FOR GRID

- **Element directives** – Directive activates when a matching element is encountered.
- **Attribute** – Directive activates when a matching attribute is encountered.
- **CSS** – Directive activates when a matching css style is encountered.
- **Comment** – Directive activates when a matching comment is encountered.

1.3 INTENDED AUDIENCE

- General Users who use the application:
 - ✓ General users will use the Grid to do specific data set implementation in page.
- UI Developers:
 - UI Developers who want to work on refining or adding new functionalities to the directive will be able to understand the basic logic and mechanisms very swiftly upon referring this document.

1.4 DEFINING A DIRECTIVE

This section lists simple steps to define a custom directive in an AngularJS module. First, we need to define an Angular app.

```
var myApp = angular.module('myApp', []);
```

Now, define a directive.

```
myApp.directive('myDirective', function() {  
  return {  
    restrict: 'E',  
    template: '<h1>I made a directive!</h1>'  
  };  
});
```

This defines a directive. restrict: 'E' means “restrict the usage of this directive to only Elements.” Thus we embed this directive in the HTML page as

```
<body ng-app="myApp">  
  <my-directive></my-directive>  
</body>
```

This code piece is equivalent to

```
<body ng-app="myApp">  
  <h1>I made a directive!</h1>  
</body>
```

Note that AngularJS maps the naming conventions from HTML's **my-directive** to JavaScript's **myDirective**

1.5 SAMPLE CODE

1.5.1 ANGULAR DIRECTIVE CODE

```
(function() {
  angular.module('ProjectNameUI.common.ProjectNameForm').directive('ProjectNameDynamicAttrGridUI', ProjectNameDynamicAttrGridUI)
  /* @ngInject */
  function ProjectNameDynamicAttrGridUI($compile, remoteCall, isUndefinedOrNullValueService, userSettings, uiGridConstants, $filter) {
    function Controller($scope, $timeout, $modal) {}
  }

  var directive = {
    restrict: 'E',
    scope: {
      configJSON: '=configjson',
      dataset: '=dataset',
      refData: '=refdata',
      checkDisable: '=?',
      multiArr: '=',
      param: '=param',
      field: '@',
      children: '=?',
      childrens: '@',
      selectedRecords: '=?',
      submitted: '=',
      gridSetting: '@gridsetting',
      serverPagination: '@serverpagination'
    },
    controller: Controller,

    link: function(scope, element, attrs, ctrl) {
      if (ctrl) {
        scope.configJSON = ctrl.configJSON;
        scope.refData = ctrl.refData;
      }
      scope.$on('page-forward', function(evt, args) {
        startServerPagination();
      });
      scope.internalControl = scope[attrs.control] || {};
      scope.columnTitle = [];
      var tableInfo = [];
      var childInfo = [];
      var tableModel = scope.configJSON[attrs.field];
      if (scope.layoutJSON) {
        tableModel = scope.layoutJSON[attrs.field];
      }
    }
  };
})
```

Angular JS Custom directive for Grid – UI Reusable component

```

/*Setting the saved grid setting ends*/

// Mongo configuration
var childEnabled = tableModel.hasOwnProperty('child') ? true : false;
var childLength = tableModel.hasOwnProperty('childlength') ? tableModel.childlength : "";

var rowDraggable = tableModel.hasOwnProperty('draggable') ? true : false;
var editEnabled = tableModel.hasOwnProperty('editable') ? tableModel.editable : !attrs.noteditable;
var filterEnabled = tableModel.hasOwnProperty('filterable') ? tableModel.filterable : !attrs.notfilterable;
var exportEnabled = tableModel.hasOwnProperty('exportable') ? tableModel.exportable : !attrs.notexportable;
var selectionEnabled = tableModel.hasOwnProperty('selectable') ? tableModel.selectable : !attrs.notselecttable;
var cellEditEnabled = tableModel.hasOwnProperty('enableCellEdit') ? tableModel.enableCellEdit : false;
var disableAllFilter = tableModel.hasOwnProperty('hidefilter') ? tableModel.hidefilter : false;
var disableDownloadOption = tableModel.hasOwnProperty('hideDownloadOption') ? tableModel.hideDownloadOption : false;
var disableSaveOption = tableModel.hasOwnProperty('hideSaveOption') ? tableModel.hideSaveOption : false;
var disableRestoreOption = tableModel.hasOwnProperty('hideRestoreOption') ? tableModel.hideRestoreOption : false;
var disableFilterOption = tableModel.hasOwnProperty('hideFilterOption') ? tableModel.hideFilterOption : false;
var disableColumnOption = tableModel.hasOwnProperty('hideColumnOption') ? tableModel.hideColumnOption : false;
var linkToCreate = tableModel.hasOwnProperty('linkToCreate') ? tableModel.linkToCreate : false;
var enableInlineEdit = tableModel.hasOwnProperty('enableInlineEdit') ? tableModel.enableInlineEdit : false;
var multiselectDisabled = tableModel.hasOwnProperty('multiselect') ? tableModel.multiselect : !attrs.notmultiselect;
var modifierKeysToMultiSelectDisabled = tableModel.hasOwnProperty('modifierKeysToMultiSelect') ? tableModel.notselec
var childmultiselectDisabled = tableModel.hasOwnProperty('childmultiselect') ? tableModel.childmultiselect : !attrs.
var childmodifierKeysToMultiSelectDisabled = tableModel.hasOwnProperty('childmodifierKeysToMultiSelect') ? tableMode
var enableServerSidePagination = tableModel.hasOwnProperty('enableServerSidePagination') ? tableModel.enableServerSi
var maxRecordsPerPage = tableModel.hasOwnProperty('maxRecordsPerPage') ? tableModel.maxRecordsPerPage : false;
var pinningEnabled = !editEnabled;
var moveEnabled = !editEnabled;
var resizeEnabled = !editEnabled;
var gridId = 'ProjectNameUiGrid_' + attrs.field + '_' + attrs.dataset;
var isRowExpanded = false;
var exporterSuppressColumns = ["actions"];

```

Angular JS Custom directive for Grid – UI Reusable component

```
function populateTable(data, height) {
    if (childEnabled) {
        paginationPageSize: parseInt(userSettings.getValNumberOfResults().property),
        data: data[i][attrs.child]
    };
    scope.gridOptions.data = data;
} else {
    angular.forEach(scope.gridOptions.columnDefs, function(value, key) {
        value1[value.name] = value1[value.name] ? moment(value1[value.name]).format(userSe
    } else if (value1[value.name] == value.displayLabel[0].code || value1[value.name] == val
        value1[value.name] = (value.displayLabel[0].code == value1[value.name]) ? value.di
    } else if (value.hasOwnProperty("displayValue")) {
    }
    });
    scope.gridOptions.data = data;
}
ctrl.gridOptions = data;
element.html(generateTemplate({
})).show();
if (!$("ProjectName-dynamic-attr-grid-ui").parent().hasClass("gridPosition")) {
    $("ProjectName-dynamic-attr-grid-ui").wrap("<div class='gridPosition'></div>");
}
$compile(element.contents())(scope);
}
// watching for dataset to update values from remote call
scope.$watch('dataset', function(dataValue) {
});
scope.configJSON.addRowHeightCalc = function (rowCount, gridIdArg, dataLength) {
};
scope.configJSON.deleteRowHeightCalc = function (rowCount, gridIdArg, dataLength) {
};
};
return directive;
```


1.5.2 HTML VIEW CODE





```
var template = '<div style="height:35px;" ng-hide="disableAllFilter"><div><ul class="tabbtn_16x16" style="padding: 5px 0 0 0;"><li><a href class="glyphicon glyphicon-filter" ng-class="{invalidcursor:disableFilterOption}" ng-disabled="disableFilterOption" tooltip="{{itemInfo.tooltip.filter}}" tooltip-trigger="mouseenter" tooltip-placement="top" ng-click="!disableFilterOption && toggleFiltering()" style="text-decoration: none; top: 1px; left: 4px;">&nbsp;</li><li><div class="hamburgerMenu dropdown toggleColumn" id="'+ gridId + '_open' style="padding: 0px; margin: 0px;"><a href class="glyphicon glyphicon-menu-hamburger dropdown-toggle" ng-disabled="disableColumnOption" ng-class="{invalidcursor:disableColumnOption}" tooltip-enable="{{itemInfo.tooltip.showTooltip}}" uib-tooltip="{{itemInfo.tooltip.options}}" tooltip-trigger="mouseenter" tooltip-placement="top" data-toggle="dropdown" aria-haspopup="true" aria-expanded="false" style="width: 40px; background: #f3f3f3; height: 24px; margin-top: 1px; border-top: 1px solid #E2E2E2; text-decoration: none; color: #2E2E2E;">&nbsp;</a><ul class="pt0 pb0 dropdown-menu" aria-labelledby="dropdownMenu1"><li ng-repeat="columnName in columnNames"><a href class="glyphicon glyphicon-ok" ng-click="hideShowColumn($index, columnName)" style="text-decoration: none; color: #2E2E2E;">&nbsp;</a></li></ul></div></li></ul></div>' +
'<li><a href class="glyphicon glyphicon-save-file" ng-click="!disableSaveOption && saveGridSetting()" ng-class="{invalidcursor:disableSaveOption}" tooltip-trigger="mouseenter" tooltip-placement="top" style="text-decoration: none; color: #2E2E2E;">&nbsp;</a></li><li><a href class="glyphicon glyphicon-repeat" ng-disabled="disableRestoreOption" ng-class="{invalidcursor:disableRestoreOption}" tooltip-enable="{{itemInfo.tooltip.showTooltip}}" uib-tooltip="{{itemInfo.tooltip.reload}}" tooltip-trigger="mouseenter" style="text-decoration: none; top: 1px; left: 4px;">&nbsp;</a></li><li><div class="dropdown hamburgerMenu" id="dropdownMenu2"><a href class="glyphicon glyphicon-save dropdown-toggle" ng-disabled="disableDownloadOption" ng-class="{invalidcursor:disableDownloadOption}" tooltip-enable="{{itemInfo.tooltip.showTooltip}}" uib-tooltip="{{itemInfo.tooltip.export}}" tooltip-trigger="mouseenter" aria-haspopup="true" aria-expanded="false" style="width: 40px; background: #f3f3f3; margin-top: 1px; height: 24px; text-decoration: none; color: #2E2E2E;">&nbsp;</a><ul style="position: absolute; left: -208px; width: 250px; top: 25px;"><li><a href ng-click="xlsExport($event)" style="text-decoration: none; color: #2E2E2E;" class="fa fa-file-excel-o">&nbsp;</a></li><li><a href ng-click="export('{{csv}}', '{{visible}}', '{{visible}}')">&nbsp;</a></li><li><a href ng-click="export('{{csv}}', '{{all}}', '{{all}}')">&nbsp;</a></li></ul></div></li></ul></div></div>';

template += '<div ui-grid="gridOptions" ui-grid-auto-resize';

template += ' id="'+ gridId + '"';
template += ' style="height:'+ params.height + 'px" ';
template += params.paginationEnabled ? ' ui-grid-pagination ' : '';
template += params.childEnabled ? ' ui-grid-expandable ' : '';
template += params.pinningEnabled ? ' ui-grid-pinning ' : '';
template += params.selectionEnabled ? ' ui-grid-selection ' : '';
template += params.cellEditEnabled ? ' ui-grid-edit ' : '';
template += params.exportEnabled ? ' ui-grid-exporter ' : '';
template += params.moveEnabled ? ' ui-grid-move-columns ' : '';
template += params.resizeEnabled ? ' ui-grid-resize-columns ' : '';
template += params.saveEnabled ? ' ui-grid-save-state ' : '';
template += params.rowDraggable ? ' ui-grid-draggable-rows ' : '';
```

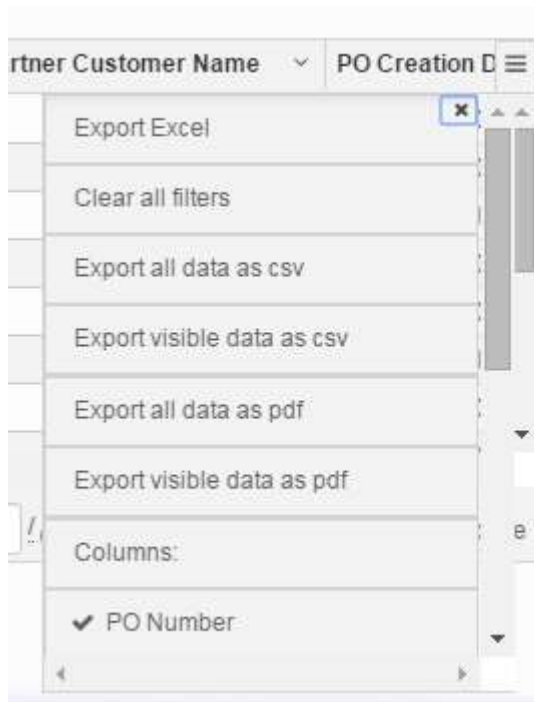
2. GRID DIRECTIVE INFORMATION USING ANGULAR JS

2.1 VIEW (SAMPLE DATA)

Actions	PO Number	Vendor	PO Status Code	Customer	Office Code	Trading Partner Customer Name	PO Creation
	TEST1	ADIDAS	Draft	BP_AGRON			2018-01-27 03
	TEST	ADIDAS	Draft	BP_AGRON			2018-01-27 03
	TEST	ADIDAS SOURCING UMI	Active	BP_AGRON			2018-01-27 10
	TEST_K	ADIDAS SOURCING UMI	Active	BP_AGRON			2018-01-27 02
	TEST_4115	ADIDAS SOURCING UMI	Active	BP_AGRON			2018-12-29 01
	TEST_1	ADIDAS SOURCING UMI	Active	BP_AGRON			2018-12-18 11
	TESTK1	ASW	Active	BP_AGRON			2018-01-29 22


1 - 15 of 76 items

14 4 1 15 Items per page



2.2 GENERAL INFORMATION ABOUT THE DIRECTIVE

- View**



Actions	PO Number	Vendor	PO Status Code	Customer	Office Code	Trading Partner Customer Name	PO Creation Date
	TST1	ADIDAS	Draft	BP_AGRON			2018-01-27
	TST	ADIDAS	Draft	BP_AGRON			2018-01-27
	Tiv1	ADIDAS SOURCING UMI	Active	BP_AGRON			2018-01-27
	TEST_K	ADIDAS SOURCING UMI	Active	BP_AGRON			2018-01-27
	TEST_1115	ADIDAS SOURCING UMI	Active	BP_AGRON			2018-12-23
	TEST_1	ADIDAS SOURCING UMI	Active	BP_AGRON			2018-12-18
	TEST1	ASW	Active	BP_AGRON			2018-01-29

- Directive Name:**
<projectName>-dynamic-attr-grid-ui
- Directive Type:**
Grid directive
- Attributes to add:**
Two attributes needs to add **configjson="configJSON"** and **dataset="searchResults"** at the time of implementation.

Ex:-

```
<projectName-dynamic- attr-grid-ui field="gridfield "
configjson="configJSON" dataset="searchResults"/>
```

- Location:**

Directive: Directive:

```
<ProjectName>_CustApps_Web_UI_Client\app\app_modules\common\projectNameForm\directives\projectNameGrid.js
```

View:

```
<ProjectName>_CustApps_Web_UI_Client\app\app_modules\common\projectNameForm\directives\views\projectNameGrid.html
```

2.3 DB JSON OBJECT STRUCTURE

```
"gridfield" : {
  "type": "grid",
  "label": "Booking Lines",
  "name": "bookingLinesConfig",
  "editable": true,
  "selectable": true,
  "enablePinning": true,
  "resizeEnabled": true,
  "actionLabel": "Action",
  "actionWidth": 30,
  "redirecttolastPage": true,
  "enableVerticalScrollbar": false,
  "enableHorizontalScrollbar": true,
  "pagingPageSizeHidden": true,
  "pagingPageSizes": [
    15
  ],
  "pagingPageSize": 15,
  "showFiltering": false,
  "dynamicheight": true,
  "elements": [{
    "label": "#",
    "width": 42,
    "pinnedLeft": true,
    "name": "bkgln_bkgLnId",
    "enableColumnResizing": false
  }, {
    "field": "bkgLnDtl_poOrderNbr",
    "poFixedFields": true,
    "enableColumnResizing": false
  }, {
    "field": "bkgLnDtl_stylCd",
    "poFixedFields": true,
    "enableColumnResizing": false
  }, {
    "field": "bkgLnDtl_subStylCd",
    "poFixedFields": true,
    "enableColumnResizing": false
  }, {
    "field": "bkgLn_otQty"
  }, {
    "field": "bkgLn_otQtyUom"
  }, {
    "field": "bkgLn_inpkQty"
  }, {
    "field": "bkgLn_inpkUom"
  }, {
    "field": "bkgLn_gohUom"
  }],
  "show": true
}
```

2.4 JSON OBJECT VARIABLE DETAILS

Variable Name	Accepts type	Explanation
type	String	Directive Type "grid"
label	String	Label for the Grid
name	String	Name for the grid
editable	Boolean	True - Inline Editing Grid. False - View Grid
selectable	Boolean	True - Selectable Grid False - Non Selectable Grid
enablePinning	Boolean	True - Allows user to freeze the column at left/right. False - Feature is not enabled.
resizeEnabled	Boolean	True - User can resize the Column Size. False - Feature is not enabled.
actionLabel	String	Label for a column which performs Actions, default is 'Action'
actionWidth	Integer	Width of the Action Column
redirecttolastPage	Boolean	True - On Page load, if grid has 10 pages, it will load the last page.

**Angular JS Custom directive for Grid –
UI Reusable component**

		False - On Page load , First page of the grid will be loaded
enableVerticalScrollbar	Boolean	True - To Enable vertical scrollbar for Grid. False - Disables the vertical scrollbar
enableHorizontalScrollbar	Boolean	True - To Enable Horizontal scrollbar for Grid. False - Disables the Horizontal scrollbar
showFiltering	Boolean	True - Displays the filter feature for each column False - Doesn't display the filter feature.
dynamicheight	Boolean	True - Changes the height according to the rows in the Grid. False - Always has a static height
show	Boolean	Display or hide
elements	String	Column to be displayed in the each rows

2.5 CURRENT ISSUES AND ADDRESSES

- All Console.log () usages have been removed from the directive.
- HTML view and the Directive have been separated.
- Unused scope variables have been removed.

2.6 PROS AND CONS

Pros:

- Easy way of creating a grid with the required functionalities with minimal HTML code.
- Re-usable component which will be used across the application.

Cons:

- Need to have some understanding in the directive code to use it to good effect.