

Angular JS Custom Directive for DualBox

(Reusable component for all the projects)

VERSION CONTROL

Prepared by:	Sankar, Meenakshi
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	Sankar, Meenakshi	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 DUALBOX.....	4
1 Introduction.....	4
1.1 Purpose	4
1.2 Supporting elements by angularjs to create directive for dropdown	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	6
2. DUALBOX INFORMATION USING ANGULAR JS.....	7
2.1 view (sample dualbox).....	7
2.2 list of buttons	7
2.3 detailed information about the 4 buttons	7
2.3.1 Add All	7
2.3.2 Add	7
2.3.3 Remove	7
2.3.4 Remove All	7
2.4 general information about the directive	8
2.5 DB JSON object structure	8
2.6 JSON object variable details	9
2.7 Current issues and Addresses	10
2.8 Pros and Cons.....	11

1. INTRODUCTION TO ANGULAR JS CUSTOM DIRECTIVE FOR DUALBOX

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.

- dualbox is a custom directive used to move list item(s) between two lists, usually a source list and a destination list.
- The source list contains available list items which can be moved into the destination list which contains the selected list items.

1.1 PURPOSE

- The dualbox is used to select and add single or multiple listed items. (subsequently, remove them, if they have been already selected)

1.2 SUPPORTING ELEMENTS BY ANGULARJS TO CREATE DIRECTIVE FOR DROPDOWN

- **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 dualbox to add or remove listed items from source lists to destination lists (also, move items in the destination list back to the available list).
- 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('projectNameDualListBox', projectNameDualListBox);
    function projectNameDualListBox($compile, $timeout, remoteCall, $filter, $parse) {

        return {
            restrict: 'E',
            scope: {
                ngModel: '=',
                configJson: '=',
                refData: '=',
                controller: '=',
                showFilter: '@',
                dualBoxScopes: '=',
                selectAll: '=',
                focus: '@'
            },
            templateUrl: 'app_modules/common/projectNameForm/directives/views/projectNameDualBox.html',
            link: function (scope, iElement, iAttrs) {

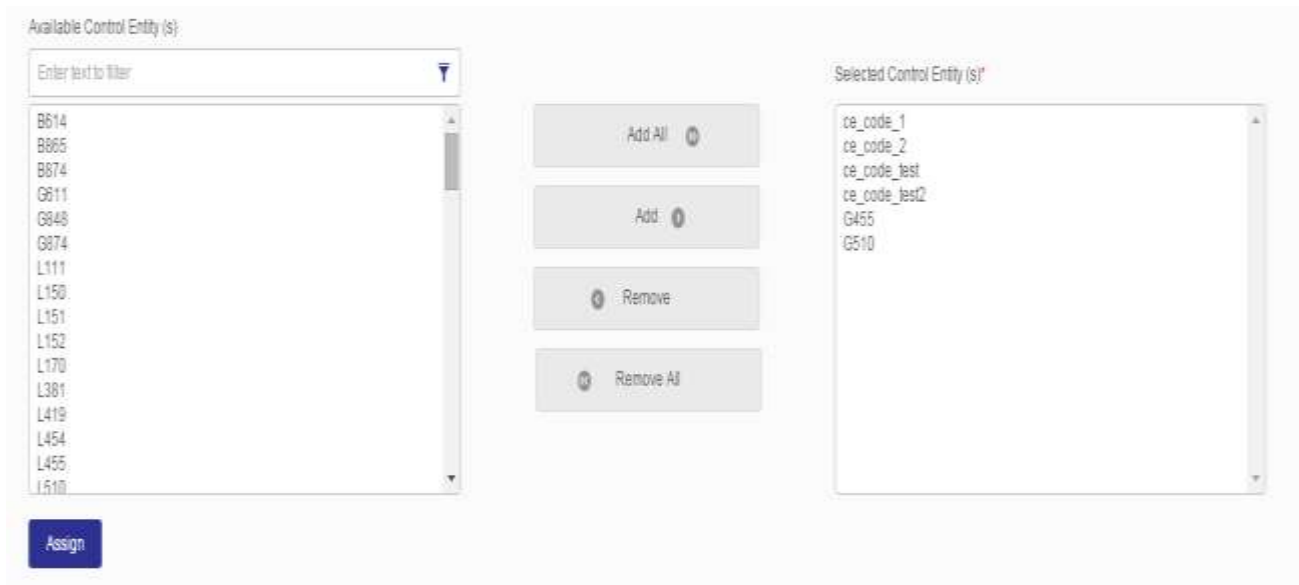
            }
        };
    }
})();
```

1.5.2 HTML VIEW CODE

```
<div ng-class="[outlinebox]">
    <table border=0 style="margin-bottom: 0px" class="dualListTable">
        <tr ng-show="showFiltering">
            <td colspan=3>{{itemInfo.filteringMessage}}</td>
        </tr>
        <tr ng-show="!showFiltering && sourceList.length>= maxRecord">
            <td colspan=3>{{itemInfo.maxLengthMessage}}</td>
        </tr>
    </table>
</div>
```

2. DUALBOX INFORMATION USING ANGULAR JS

2.1 VIEW (SAMPLE DUALBOX)



2.2 LIST OF BUTTONS

- Add All
- Add
- Remove
- Remove All

2.3 DETAILED INFORMATION ABOUT THE 4 BUTTONS

2.3.1 Add All

Add all the list items from the box on the left to the box on the right.

2.3.2 Add

Add the selected list item(s) from the left box to the box on the right.

2.3.3 Remove

Remove the selected list items(s) from the box on the right and move it to the box on the left.

2.3.4 Remove All

Remove all the list items(s) from the box on the right and move it to the box on the left.

2.4 GENERAL INFORMATION ABOUT THE DIRECTIVE

- **Directive Name:**
projectName-duallistbox
- **Directive Type:**
Element Level Directive
- **Attributes:**
field, ng-model, displayfield, config-json, ref-data, placeholder, outlineBox, enableremotecall
- **Location:**
Directive:
<projectName>_CustApps_Web_UI_Client\app\app_modules\common\ProjectNameForm\directives\projectNameDualBox.js
View:
<projectName>_CustApps_Web_UI_Client\app\app_modules\common\projectNameForm\directives\views\projectNameDualBox.html

2.5 DB JSON OBJECT STRUCTURE

```
{
  "dual_customers" : {
    "type" : "duallist",
    "name" : "",
    "label" : "",
    "showLabel" : true,
    "show" : true,
    "default" : "",
    "showSelectAllOpt" : true,
    "selectAllOptLabel" : "",
    "disabled" : false,
    "labelDest" : "",
    "filteringMessage": "Retrieving values...",
    "maxLengthMessage": "The displayed list is not comprehensive. Use the text box filter to refine the list.",
    "maxRecord": 499,
    "options" : {
      "dataSource" : "",
      "value" : "",
      "label" : "",
      "typeCd" : "",
      'custom' : '',
      'url' : ''
    }
  }
}
```


2.6 JSON OBJECT VARIABLE DETAILS

Variable Name	Accepts type	Explanation
type	String	Directive Type "duallist"
name	String	
showLabel	Boolean	Display or hide label
show	Boolean	Display or hide directive
default	String	
showSelectAllOpt	Boolean	Display or hide checkbox to enable select All option. Checking this box will move all source list items to destination list without having the user click on the "add all" button from inside the dual list box.
selectAllOptLabel	String	Label for the select all option checkbox
disabled	Boolean	To set ng-disabled attribute value
labelDest	String	Destination list label
dataSource	String	Data source for the source list
value	String	

label	String	
typeCd	String	
custom	String	
url	String	URL for RemoteCall
filteringMessage	String	Message to be displayed when Filtering Textbox is being used to filter source list items
maxLengthMessage	String	Message to be displayed when the list of items displayed in the listbox is less than the actual number of items on the list.
maxRecord	Int	maxRecord to set limit for number of items displayed in the duallistbox.

2.7 CURRENT ISSUES AND ADDRESSES

- Static Messages have been moved to DB so that they can be customized and delivered with more specificity.
- The maximum number of list items that can be displayed in the directive was previously hardcoded, now that value has been moved to DB and can be set individually for each page and for each page level implementation of dualbox by editing it in the DB configJSON.
- All Console.log() usages have been removed from the directive.
- HTML view and the Directive have been separated.
- Unused scope variables have been removed.

2.8 PROS AND CONS

Pros:

- Offers a very intuitive and easy way to add items from a list of available items to a list of selected items
- The filtering box offers a quick way to filter extensive lists, to quickly find the desired list item a through simple search mechanism

Cons:

- If the applied Displayfield is not present in the object returned by the data API, the dualbox will be empty.