

Angular JS custom Tooltip with Multi- language support (UIB)

VERSION CONTROL

Prepared by:	Dhivya Arinathan
Date:	22/01/2016
Reviewed and Accepted by:	Monesh Pattar, B.C.Subramanian, S.Kalaiyarasu
Date:	12/02/2016
Approved by:	Baskaran Varadarajan, Soundararajan Das
Date:	13/02/2016

VERSION HISTORY

Date	Version	Author	Description
22/01/2016	0.1	Dhivya Arinathan	Initial draft
30/01/2016	1.1	Monesh Pattar	First version
05/02/2016	1.2	B.C.Subramanian	Second Review
10/02/2016	1.3	S.Kalaiyarasu	Third Review
13/02/2016	2.0	Baskaran Varadarajan	Final Approved

Contents

1. ANGULAR JS CUSTOM TOOLTIP WITH MULTI-LANGUAGE SUPPORT (UIB)	4
1.1 Tooltip advantages using angular js.....	4
1.2 Angular js tooltip features.....	5
1.3 Angular version to implement tooltip (UIB).....	5
2. ANGULAR JS TOOLTIP IMPLEMENTATION STEP BY STEP APPROACH.....	7
2.1 Angular custom directive steps to configure tooltip with view component:	7
2.2 DB configuration details to avoid static strings:	7
2.3 Tooltip html view code steps:	8
3. ANGULAR JS TOOLTIP - SAMPLE VIEW	9
3.1 Static tooltip sample view:	9
3.2 Dynamic tooltip sample view:	9
3.3 Button tooltip:	9
3.4 Checkbox tooltip:	10
3.5 Dropdown:	10
3.6 Grid tooltip:	10
4. ANGULAR JS TOOLTIP CONFIGURATION FOR MULTI-LANGUAGE.....	11
4.1 Sample to change the language to provide multi-language support:.....	11
4.2 Sample view for the Multi-language support:	11

1. ANGULAR JS CUSTOM TOOLTIP WITH MULTI-LANGUAGE SUPPORT (UIB)

Tooltip is a common graphical user interface element. It is used in conjunction with a cursor, usually a pointer. The user hovers the pointer over an item, without clicking it, and a tooltip may appear—a small "hover box" with information about the item being hovered over.

- A tooltip can be used to reveal more info about a control before asking the user to perform an action.
- Tooltip makes interface more intuitive.
- The tooltip is visible when the button is hovered, focused, or touched.

1.1 TOOLTIP ADVANTAGES USING ANGULAR JS

- The tooltip attributes that you place on the element are directives themselves, so the tooltip needs to be compiled when it's appended. That's not a problem, you can use the \$compile service to do this, but you need to do it just once for the element.
- Also, you need to bind the tooltip text (the value given to the tooltip attribute) to an expression. I do that by passing in a concatenated value of \$interpolate.startSymbol() + the scope value that you want to display (in the demo it is the fieldx property of the errors object) + the \$interpolate.endSymbol(). This basically evaluates to something like: {{error.field1}}. I use the \$interpolate service start and end symbols because it just makes the directive more componentized, so you can use it on other projects where you might have multiple frameworks and be using something other than double curly-braces for your Angular expressions. It's not necessary though and you could instead do: '{{'+errorObj+'.'+inputName+'}}'. In this case, you don't have to add the \$interpolate service as a dependency.
- As you can see, to make the directive truly reuseable, rather than hard-coding the error field, I set the value given to the directive attribute to the name of the object that will be watched and use the input name value as the object property.
- The chief thing you need to remember is that before you compile, **you have to remove the error-tooltip attribute** from the element because if you don't you'll wind up in an infinite loop and crash hard! Basically, the compile service is going to take the element that the directive is attached to and compile it with all of the attributes your directive added, if you leave the error-tooltip attribute, it's going to try and recompile that directive too.
- Lastly, you can take advantage of the fact that the tooltip will not display if its text value is empty or undefined. That means you only have to watch the errors object not the individual property on the error associated with the tooltip.

1.2 ANGULAR JS TOOLTIP FEATURES

- Tooltip supports multiple placements as mentioned below,
 - top - tooltip on top, horizontally centered on host element.
 - top-left - tooltip on top, left edge aligned with host element left edge.
 - top-right - tooltip on top, right edge aligned with host element right edge.
 - bottom - tooltip on bottom, horizontally centered on host element.
 - bottom-left - tooltip on bottom, left edge aligned with host element left edge.
 - bottom-right - tooltip on bottom, right edge aligned with host element right edge.
 - left - tooltip on left, vertically centered on host element.
 - left-top - tooltip on left, top edge aligned with host element top edge.
 - left-bottom - tooltip on left, bottom edge aligned with host element bottom edge.
 - right - tooltip on right, vertically centered on host element.
 - right-top - tooltip on right, top edge aligned with host element top edge.
 - right-bottom - tooltip on right, bottom edge aligned with host element bottom edge.
- Supports various triggers like mouse enter, click, outside Click and focus.
- It can be enabled and disabled by setting true or false respectively to the tooltip-enable attribute.
- To provide internationalization, all the values can be configured in DB Script file.
- The tooltip can display static messages defined in the DB Script file or HTML. In addition to this it can also display dynamic values like when a user enters characters in a textbox, the characters entered by the user can also be displayed in the tooltip.

1.3 ANGULAR VERSION TO IMPLEMENT TOOLTIP (UIB)

This repository contains a set of **native AngularJS directives** based on Bootstrap's markup and CSS. As a result no dependency on jQuery or Bootstrap's JavaScript is required. The **only required dependencies** are:

- AngularJS (requires AngularJS 1.4.x, tested with 1.4.9). 0.14.3 is the last version of this library that supports AngularJS 1.3.x and 0.12.0 is the last version that supports AngularJS 1.2.x.
- Angular-animate (the version should match with your angular's, tested with 1.4.9) if you plan in using animations, you need to load angular-animate as well.

Bootstrap CSS (tested with version 3.3.6). This version of the library (1.1.2) works only with Bootstrap CSS in version 3.x. 0.8.0 is the last version of this library that supports Bootstrap CSS in version 2.3.x.

	Version
AngularJS	1.3.x
Angular Bootstrap	0.14.3

2. ANGULAR JS TOOLTIP IMPLEMENTATION STEP BY STEP APPROACH

To implement Tooltip using angular js, we have followed three sections of concepts.

1. Directive configuration to reuse the same tooltip concept for all the view and controller pages.
2. DB section to get contents dynamically and one time changes if required.
3. View pages to render and apply tooltip dynamic contents to display.

2.1 ANGULAR CUSTOM DIRECTIVE STEPS TO CONFIGURE TOOLTIP WITH VIEW COMPONENT:

```
if (scope.itemInfo.tooltip && scope.itemInfo.tooltip.showTooltip &&
scope.itemInfo.tooltip.message)
{
    scope.displayTooltip = scope.itemInfo.tooltip.message;
}
if (scope.itemInfo.tooltip && scope.itemInfo.tooltip.message &&
scope.itemInfo.tooltip.showTooltip){
    scope.itemInfo.tooltip.placement =
(scope.itemInfo.tooltip.placement)?scope.itemInfo.tooltip.placement:'top';
}
```

Based on the field that we need, will get string information from MongoDB and pass the same to view page through directive logic as mentioned above.

In the above implementation, the changes and information will be stored in the scope.itemInfo.tooltip.placement field based on conditions.

2.2 DB CONFIGURATION DETAILS TO AVOID STATIC STRINGS:

Through this DB configuration, the string value will be fetched from Db and can be managed dynamically.

```
"tooltip" : {
    "showTooltip" : true,
    "message" : 'Click the button to search for records',
    "placement" : 'top'
}
```

This configuration will be working like resource file. We have maintained the contents as key value paired and the same will be manipulated via tooltip directive code.

In the sample code above, all the key value contents are enclosed within tooltip reference name.

2.3 TOOLTIP HTML VIEW CODE STEPS:

```
<button ng-class="itemInfo.class" name="{{attrName}}" ng-disabled="condition"
ng-model="model[itemInfo.name]"
  tooltip-enable="{{itemInfo.tooltip.showTooltip}}"
  uib-tooltip="{{displayTooltip}}"
  tooltip-trigger="mouseenter"
  tooltip-placement="{{itemInfo.tooltip.placement}}"
  ProjectName-auto-focus="{{focusFirstInput}}"
  ng-click="clickFunc()" ng-show="definedButtonShow">{{itemInfo.label}}
</button>
```

We have to map this tooltip to view page through directive code. For example, enabling the tooltip for one of the button directive as mentioned above. So whenever hover the mouse on this button in the view page, will be getting the tooltip message as configured in DB.

3. ANGULAR JS TOOLTIP - SAMPLE VIEW

3.1 STATIC TOOLTIP SAMPLE VIEW:

The screenshot shows a search form titled "Search Code-Decode Profile". It includes a text input field with a placeholder "Enter customer name or code". A static tooltip is displayed above this field with the text "Enter customer name or code". Other fields include "Process" (a dropdown menu showing "PO Intern Batch Job - P"), "External System" (a dropdown menu), "Attribute" (a text input field), and radio buttons for "Inbound", "Both", and "Outbound". Below these fields is an "Advanced Search" section with a "Search" button and a "Reset Fields" button.

3.2 DYNAMIC TOOLTIP SAMPLE VIEW:

Dynamic tooltip will be displayed, whenever we type something in the input field.

The screenshot shows the same search form as in 3.1, but with a dynamic tooltip. The text input field now contains the text "AGRON INC - (DELTA) - BI". The tooltip is updated to show "AGRON INC - (DELTA) - BI". The rest of the form, including the "Process" dropdown, "External System" dropdown, "Attribute" text input, and "Advanced Search" section, remains the same.

3.3 BUTTON TOOLTIP:

The screenshot shows the search form with a tooltip pointing to the "Search" button. The tooltip text is "Click the button to search for records". The input field contains the text "AGRON INC - (DELTA) - BI". The rest of the form, including the "Process" dropdown, "External System" dropdown, "Attribute" text input, and "Advanced Search" section, remains the same.

3.4 CHECKBOX TOOLTIP:

The screenshot shows a search form titled "Search Code-Decode Profile". It includes fields for Customer (with a dropdown), Process (with a dropdown), External System (with a dropdown), and Attribute (with a text input). There are checkboxes for "Both", "Inbound", and "Outbound". A tooltip is displayed over the "Both" checkbox, containing the text "Check this box to apply to both". Below the form is an "Advanced Search" button.

3.5 DROPDOWN:

The screenshot shows the same search form as in 3.4. A tooltip is displayed over the "Process" dropdown menu, containing the text "Choose a process". Below the form are "Search" and "Reset Fields" buttons.

3.6 GRID TOOLTIP:

The screenshot shows a data grid with columns: Actions, Sender ID, Receiver ID, Message Type, and Status. A tooltip is displayed over the "Geodis Wilson - ZZGWHQMB01" entry in the Sender ID column.

Actions	Sender ID	Receiver ID	Message Type	Status
	Yusen Logistics East - 02YYKN	APLUNET	INBOUND TRUCKER (214T)	Active
	Yusen Logistics West - 02YVWC	APLUNET	INBOUND TRUCKER (214T)	Active
	Geodis Wilson - ZZGWHQMB01	APLUNET	INBOUND TRUCKER (214T)	Active
	Geodis Wilson - ZZGWHQMB01	TEST	INBOUND OCEAN (315 INBOU	Active
	Ziegler Logistics (Hong Kong) Ltd - ZZDEQLERPROG	APLUNET	INBOUND OCEAN (315 INBOU	Active
	NYCE - 02NYCI	APLUNET	INBOUND TRUCKER (214T)	Active
	Great Central Transport - 02GCTN	APLUNET	INBOUND TRUCKER (214T)	Active
	W&B Trucking - 02WBKG	APLUNET	INBOUND TRUCKER (214T)	Active
	DECON Extended - COI - ZZDECON_TOA_EXT	APLUNET	INBOUND TRUCKER (214T)	Active
	eF3 Systems Inc - ZZEF3SY99	APLUNET	INBOUND TRUCKER (214T)	Active
	Schenker of Canada Ltd - ZZSICLTO	APLUNET	INBOUND OCEAN (315 INBOU	Active
	Roadline Intermodal Logistic - 02RDKA	APLUNET	INBOUND TRUCKER (214T)	Active
	Container Connection - 02CNCS	APL	INBOUND OCEAN (315 INBOU	Active
	Kontabou World Express - ZZKINGLBUSP	APLUNET	INBOUND OCEAN (315 INBOU	Active
	CalCartage Express - 02CCKP	APLUNET	INBOUND AIR (214A)	Inactive

16 - 30 of 260 items

4. ANGULAR JS TOOLTIP CONFIGURATION FOR MULTI-LANGUAGE

Multi-language support has implemented through MongoDB by setting value to specific key to the different language.

4.1 SAMPLE TO CHANGE THE LANGUAGE TO PROVIDE MULTI-LANGUAGE SUPPORT:

```
"Search" : {
  "type" : "button",
  "label" : "Search",
  "name" : "Search",
  "class" : "btn_primary",
  "show" : true,
  "link" : "Search(dataJSON, '/common/searchCodeDecode')",
  "tooltip" : {
    "showTooltip" : true,
    "message" : "Cliquez sur le bouton pour rechercher des enregistrements",
    "placement" : 'top'
  }
},
```

4.2 SAMPLE VIEW FOR THE MULTI-LANGUAGE SUPPORT:

French

Spanish: