Contents

[Directives 2](#_Toc420314015)

[vr-form 2](#_Toc420314016)

[vr-select: 2](#_Toc420314017)

[vr-textbox 3](#_Toc420314018)

[vr-datetimepicker 3](#_Toc420314019)

[vr-switch 3](#_Toc420314020)

[vr-button 3](#_Toc420314021)

[vr-actionbar 4](#_Toc420314022)

[vr-panel 4](#_Toc420314023)

[vr-section 4](#_Toc420314024)

[vr-row 4](#_Toc420314025)

[vr-columns 4](#_Toc420314026)

[vr-datagrid 4](#_Toc420314027)

[vr-datagridcolumn 5](#_Toc420314028)

[vr-datagrid API 5](#_Toc420314029)

[Services 6](#_Toc420314030)

[VRNavigationService 6](#_Toc420314031)

[VRNotificationService 6](#_Toc420314032)

[VRModalService 7](#_Toc420314033)

# Directives

## vr-form

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Default Value | Notes |
| name | The name of the form |  |  |

## vr-select:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Default Value | Notes |
| label | The display text |  |  |
| entityname | The name of the entity (e.g., Carrier Account, Zone…) |  |  |
| datasource | Two possible values:   * Array * function(filter): it should return promise in this case |  |  |
| ismultipleselection | true/false | false |  |
| selectedvalues | Array: in case of multiple selection.  Object: in case of single selection |  |  |
| hideselectedvaluessection | true/false | false |  |
| datatextfield/datavaluefield | The properties to use from the data source items to display and identify the item. |  | If these attributes are not specified, the item itself will be used from the datasource |
| onselectionchanged | =function()  Occurs whenever the user select/deselect an item |  |  |
| hideFilterBox | True/false | False | Hide the text box the allow the user to filter the data source list |
| isrequired | true/false | false |  |
| validator | =function()  Used for custom validation. Should return true or false |  |  |

## vr-textbox

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Default Value | Notes |
| label | The display text |  |  |
| value | Text value |  |  |
| onvaluechanged | =function() |  |  |
| isrequired |  | false |  |
| customvalidate | =function()  Used for custom validation. Should return true or false |  |  |

## vr-datetimepicker

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Default Value | Notes |
| label | The display text |  |  |
| type | date/datetime |  |  |
| value | Of type Date |  |  |
| onvaluechanged | =function() |  |  |
| isrequired | True/false | false |  |
| customvalidate | =function()  Used for custom validation. Should return true or false |  |  |

## vr-switch

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Default Value | Notes |
| Label | The display text |  |  |
| value | True/false |  |  |
| onvaluechanged | =function() |  |  |

## vr-button

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Default Value | Notes |
| type | Add, Edit, Delete, Save, Cancel, Clear, Print, Export |  |  |
| onclick | =function().  This function should return a promise in case it is calling a WebAPI. This way the button shows a loader until the response is returned from the WebAPI |  |  |
| formname | The name of the form to validate before enabling the button |  |  |

## vr-actionbar

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Default Value | Notes |
| actions | Array of actions each of which has the same properties as the vr-button directive |  |  |
| showcollapsebutton | True/false. Shows the button to expand/collapse another section | False |  |
| issectioncollapsed | True/false. Used to show/hide the section when the user clicks the expand/collapse button. showcollapsebutton should be equal to true |  |  |

## vr-panel

## vr-section

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Default Value | Notes |
| title | The title of the section |  |  |

## vr-row

## vr-columns

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Default Value | Notes |
| colnum | Number of columns |  |  |

## vr-datagrid

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Default Value | Notes |
| datasource | Array: data items |  |  |
| onReady | =function(api). Check “vr-datagrid API” section |  |  |
| pagersettings | In case if paging is needed. Example:  { currentPage: 1, totalDataCount: 0, pageChanged: function () { getData();  }  } |  |  |
| maxheight | Example: maxheight="600px" |  |  |
| hideheader |  |  |  |
| loadmoredata | =function(). It is called whenever the user scrolls down in order to retrieve more items and add them to the datasource. It should return a promise |  |  |

The vr-datagrid should be at least used with the vr-datagridcolumn directive

Example:

<vr-datagrid datasource="data" maxheight="600px">

<vr-datagridcolumn headertext="'Time'" field="'dateTimeValue'"></vr-datagridcolumn>

<vr-datagridcolumn ng-repeat="measureType in measureTypes" headertext="measureType.description" field="'Values[' + $index + ']'" type="'Number'"></vr-datagridcolumn>

</vr-datagrid>

### vr-datagridcolumn

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Description | Default Value | Notes |
| field | The property to bind to in the datasource |  |  |
| headertext | The header text |  |  |
| isclickable | Boolean Function or value indicating whether the cell is clickable. The function should be: =function(dataItem) |  |  |
| onclicked | =function(colDef, dataItem). It is called when a cell is clicked |  |  |
| type | The data type. Can be ‘Number’ |  |  |
| onsortchanged | = (colDef, sortDirection). It is called when the user clicks on a column to sort the data. It should return a promise in case it is calling a WebAPI |  |  |

### vr-datagrid API

the following table shows the members of the datagrid directive API (obtained by attaching to the on-ready event of the directive – see first table):

|  |  |
| --- | --- |
| Method | Description |
| reset() | Called to clear the user actions on the datagrid. (e.g. column sorting) |
| itemAdded(item), itemUpdated(item), itemDeleted(item) | Respective method should be called whenever an item is added, updated, or deleted. For example, when the user adds a Route Rule, the itemAdded method should be called on the datagrid API. This will shows another section to allow the user to track his actions |

# Services

## VRNavigationService

|  |  |  |
| --- | --- | --- |
| Method Name | Description | Example Usage |
| goto(url, parameters) | It navigates to specified URL. The parameters should be an object | var parameters = {  zoneId: zoneItem.EntityId,  zoneName: zoneItem.EntityName  };  VRNavigationService.goto("/BI/ZoneDetails", parameters); |
| getParameters(scope) | Retrieves the parameters sent to the page (when navigating to the page or when opening the page as a modal). the returned value is an object | var parameters = VRNavigationService.getParameters($scope);  zoneId = parameters.zoneId;  $scope.zoneName = parameters.zoneName; |

## VRNotificationService

|  |  |  |
| --- | --- | --- |
| Method Name | Description | Example Usage |
| showInformation(message) | It notifies user with information message |  |
| showSuccess(message) | It notifies user with success message |  |
| showError(message) | It notifies user with error message |  |
| showWarning(message) | It notifies user with warning message |  |
| showConfirmation(message) | It shows confirmation dialog. It returns a promise | VRNotificationService.showConfirmation('Are you sure you want to delete the Rule?').then(function (result) {  if (result) console.log('Confirmed');  else console.log('not confirmed');  }); |
| notifyException(error) | It should be called to notify the user when an exception occured |  |
| notifyExceptionWithClose(error) | It should be called in case an exception occurred when the user opens a new page (e.g. on retrieving data from the backend). It closes the page and navigates the user to an error page |  |
| notifyOnItemAdded(itemType, insertOperationOutput) | It should be called to notify the user about the result of Add operation:   * itemType: e.g. “User”, “Route Rule”… * insertOperationOutput: the output returned from the WebAPI of type InsertOperationOutput |  |
| notifyOnItemUpdated(itemType, updateOperationOutput) | It should be called to notify the user about the result of Update operation:   * itemType: e.g. “User”, “Route Rule”…   updateOperationOutput: the output returned from the WebAPI of type UpdateOperationOutput |  |

## VRModalService

|  |  |
| --- | --- |
| Method Name | Description |
| showModal(viewUrl, parameters, settings) | It opens the specified URL in a popup.   * viewUrl: the URL of the page to open * parameters: the parameters to pass to the page in the modal. it should be an object. And it could be retrieved using the getParameters(scope) method of the VRNavigationService * settings: an object having the following properties:   + useModalTemplate: true/false. Should be true If the page html is not built using the Modal directives (check next section)   + width: the width of the modal (e.g. “80%”)   + maxHeight: the height after which the modal shows a vertical scroll (e.g. “800px”)   + onScopeReady: should be equal to function(modalScope). Can be used as a communication context between the page and the modal. here are modalScope available methods:     - modalContext.closeModal(): it closes the modal |

Two options exists to be able to open a page as a modal:

1. build the html page which will open in a modal using the modal directives
2. set settings.useModalTemplate to true when calling the showModal method

Two directives exist of the modal:

|  |  |  |
| --- | --- | --- |
| Directive | Restrict | Notes |
| vr-modalcontent | Attribute Directive | Should be set on the root html element |
| vr-modalbody | Element Directive | This directive should encapsulate the main content in the page |

Example:

<div vr-modalcontent>

<vr-modalbody maxheight="'500px'">

//Main Content Here

//Main Content Here

//Main Content Here

//Main Content Here

</vr-modalbody>

<div>

//ACTION BAR HERE

//ACTION BAR HERE

//ACTION BAR HERE

//ACTION BAR HERE

</div>

</div>