**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AngJsDemoBuiltInDirec\_001.html\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html ng-app="directivesApp" >

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Built In Directive Demo</title>

<style type="text/css">

.customClass{

color:blue;

font-family:candara;

font-size:20pt;

text-transform:uppercase;

letter-spacing:1cm;

}

.evenElementStyle{

float:left;

color:white;

background-color:green;

font-family : candara;

}

.oddElementStyle{

float:left;

color:white;

background-color:crimson;

font-family : candara;

}

</style>

</head>

<body bgcolor="pink">

<!-- using $rootScope property -->

<div>

<span><b>Using $rootScope :</b>

<h2>{{compNameGlobalProp}}</h2>

</div>

<hr/>

<!-- ngBind Directive -->

<div data-ng-controller="DirectivesController">

<span><b>Using Expression :</b></span>

<span>{{"My Comp Location Is :"}}{{companyLocation}}</span>

</div>

<hr/>

<div data-ng-controller="DirectivesController">

<span><b>Using ngBind :</b> </span>

<span data-ng-Bind="companyLocation"></span>

</div>

<hr/>

<!-- ngBindTemplate Directive -->

<div data-ng-controller="DirectivesController">

<span><b>Using ngBindTemplate :</b></span>

<span data-ng-bind-template="{{'Company Location :'+companyLocation}}"></span>

</div>

<hr/>

<!-- ngBindHtml Directive Error :

"Attempting to use an unsafe value in a safe context" if ngSanitize not included as module dependency.-->

<div data-ng-controller="DirectivesController">

<span><b>Using ngBindHtml without $sce service:</b></span>

<span data-ng-bind-html="unsafeContent"></span>

</div>

<div data-ng-controller="DirectivesController">

<span><b>Using ngBindHtml with $sce service:</b></span>

<span data-ng-bind-html="safeContent"></span>

</div>

<hr/>

<!-- ngStyle Directive-->

<div data-ng-controller="DirectivesController">

<span><b>Using ngStyle:</b></span>

<span data-ng-style="customStyle">Custom Style applied.</span>

<pre>Custom style given : {{customStyle}}</pre>

</div>

<hr/>

<!-- ngClass Directive-->

<div data-ng-controller="DirectivesController">

<input type="button" value="Set Style"

data-ng-click="cssClass='customClass'"/>

<input type="button" value="Clear Style"

data-ng-click="cssClass=''"/>

</div>

<div data-ng-controller="DirectivesController">

<span><b>Using ngClass:</b></span>

<span data-ng-class="cssClass">igate</span>

</div>

<hr/>

<!-- ngRepeat, ngClassEven & ngClassOdd Directive-->

<div data-ng-controller="DirectivesController">

<span><b>Using ngRepeat, ngClassEven &amp;

ngClassOdd Directives</b></span>

<ul id="names" data-ng-repeat="name in names">

<li data-ng-class-even="cssEvenClass"

data-ng-class-odd="cssOddClass">{{name}}</li></br>

</ul>

</div>

<br style="clear:left"/>

<hr/>

<!-- ngHref & ngSrc Directive-->

<div data-ng-controller="DirectivesController">

<span><b>Using ngSrc - Adding Image: </b></span>

<image data-ng-src="{{'images/'+imageSource}}"

alt="imageSource" title="{{imageSource}}"/>

</div>

<div data-ng-controller="DirectivesController">

<span><b>Using ngHref - Adding URL: </b></span>

<a data-ng-href="{{'https://'+urlSource}}">{{urlSource}}</a>

</div>

<hr/>

<!-- ngDisabled, ngChecked, ngReadonly & ngSelected -->

<div>

<span><b>Using ngDisabled:</b></span>

<input type="button" data-ng-disabled="true" value="Disabled button"/>&nbsp;&nbsp;</br>

<input type="button" data-ng-disabled="true" value="Cancle" />&nbsp;&nbsp;</br>

<span><b>Using ngChecked:</b></span>

<input type="checkbox" data-ng-checked="true" />&nbsp;&nbsp;</br>

<span><b>I Have:</b></span>

<input type="checkbox" name="chkVehicle" value="Car" data-ng-checked="false" >Car</input>&nbsp;&nbsp;

<input type="checkbox" name="chkVehicle" value="Bike" data-ng-checked="true" >Bike</input>&nbsp;&nbsp;

<input type="checkbox" name="chkVehicle" value="Bi-Cycle" data-ng-checked="false" >Bi-Cycle</input>&nbsp;&nbsp;

<span><b>Using ngReadonly:</b></span>

<input type="textbox" value="Read only textbox" data-ng-readonly="true" />&nbsp;&nbsp;</br>

<input type="textbox" name="txtCompName" value="IGate Globle Sol Ltd " data-ng-readonly="true" />&nbsp;&nbsp;

<span><b>Using ngSelected:</b></span>

<select>

<option value="Banglore">Bangalore</option>

<option value="Pune">Pune</option>

<option value="Noida">Noida</option>

<option data-ng-selected="true" value="Chennai">Chennai</option>

</select>

</div>

<hr/>

<!-- ngIf Directive-->

<div>

<span><b>Using ngIf : </b></span>

<span data-ng-if="2 + 2 == 5">This DOM node wont be available in DOM</span>

<span data-ng-if="2 + 2 == 4">If condition is true. That's why I am getting displayed</span>

</div>

<hr/>

<!-- ngModel Directive-->

<div>

<span><b>Using ngModel : </b></span>

<input type="text" data-ng-model="sampleText"/>

</div>

<div>

<span><b>Text typed : </b></span>

<span data-ng-bind="sampleText"></span>

</div>

<hr/>

<div data-ng-controller="DirectivesController">

<!-- ngSwitch Directive -->

<div >

<span><b>Using ngSwitch:</b></span>

<select data-ng-model="selection" data-ng-options="name for name in names">

<option value="">Select a Name</option>

</select>

</div>

<div data-ng-switch on="selection" >

<div data-ng-switch-when="Latha Subramanian">You have selected Latha Subramanian</div>

<div data-ng-switch-when="Karthik Muthukrishnan">You have selected Karthik Muthukrishnan</div>

<div data-ng-switch-default>You have not selected Karthik Muthukrishnan / Latha Subramanian</div>

</div>

</div>

<hr/>

<!-- ngInclude Directive -->

<div data-ng-controller="DirectivesController">

<span><b>Using ngInclude:</b></span>

<div data-ng-include="partialPage"></div>

</div>

<hr/>

</body>

<script type="text/javascript" src="Scripts/angular.js"></script>

<script type="text/javascript" src="Scripts/angular-sanitize.js"></script>

<script type="text/javascript" >

var directiveNameSpace = angular.module('directivesApp',['ngSanitize']);

directiveNameSpace.run(function($rootScope) {

$rootScope.compNameGlobalProp = "You are creating property on global Scope My Comp name IGate";

});

directiveNameSpace.controller('DirectivesController',function($scope,$sce){

$scope.companyLocation = "IGATE Hinjavadi Pune";

$scope.unsafeContent =

"<span style='color:red'>Content won't appear in red</span>";

$scope.safeContent =

$sce.trustAsHtml("<span style='color:red'>Content should appear in red</span>");

$scope.customStyle = {'color':'yellow','background-color':'black','font-family':'candara'};

$scope.cssClass = "customClass";

$scope.cssEvenClass="evenElementStyle";

$scope.cssOddClass="oddElementStyle";

$scope.names = [

"Harish Kasthurirangan",

"Hiranmai Ramakrishnan",

"Karthik Muthukrishnan",

"Latha Subramanian",

"Mangala Rangaswamy",

"Rashmi Keshavamurthy",

"Sarbananda Behera",

"Sathiabama Ranganathan",

"Tanmaya Acharya",

"Uma Maheshwari",

"Veena Keshavalu",

"Vinod Manoharan"

];

$scope.imageSource = "delete.png";

$scope.urlSource = "www.google.com";

$scope.partialPage = "partials/partial.html";

});

</script>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AngJsDemoBuitInEventsDirective\_002.html\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<html ng-app="eventApplication">

<head>

<title>AngularJS - Event Directives</title>

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />

</head>

<body bgcolor="cyan">

<h1>AngularJS - Event Directives</h1>

<div ng-controller="EventController">

<h3>ngClick</h3>

<hr/>

<input type="button" ng-click="IncrementCounter()" value="Click to increment the counter"/><br/>

<div>

<span>Counter : {{counter}}</span>

</div>

<h3>ngMouseup</h3>

<hr/>

<input type="button" ng-mouseup="IncrementCounter()" value="Press and release mouse left button to increment the counter"/><br/>

<div>

<span>Counter : {{counter}}</span>

</div>

<h3>ngMousedown</h3>

<hr/>

<input type="button" ng-mousedown="IncrementCounter()" value="Press mouse left button to increment the counter"/><br/>

<div>

<span>Counter : {{counter}}</span>

</div>

<h3>ngMousemove</h3>

<hr/>

<input type="button" ng-mousemove="IncrementCounter()" value="Increment the counter when mouse pointer moves on the button"/><br/>

<div>

<span>Counter : {{counter}}</span>

</div>

<h3>ngMouseover</h3>

<hr/>

<input type="button" ng-mouseover="IncrementCounter()" value="Increment the counter when mouse pointer moves over the button"/><br/>

<div>

<span>Counter : {{counter}}</span>

</div>

<h3>ngMouseenter &amp; ngMouseLeave</h3>

<hr/>

<input type="button" ng-mouseenter="ChangeCounter(true)" ng-mouseleave="ChangeCounter(false)" value="Increment the counter while mouse enters and decrement when mouse leaves"/><br/>

<div>

<span>Counter : {{counter}}</span>

</div>

<h3>ngChange</h3>

<hr/>

<div>

<input type="checkbox" ng-change="ChangeStatus(status)" ng-model="status" />

<input type="button" value="{{statusText}}"/><br/>

</div>

</div>

</body>

<script type="text/javascript" src="Scripts/angular.js"></script>

<script type="text/javascript" >

var eventNameSpace = angular.module('eventApplication',[]);

eventNameSpace.controller('EventController',function($scope){

$scope.counter = 0;

$scope.status=false;

$scope.statusText="Unchecked";

$scope.IncrementCounter = function(){

$scope.counter++;

};

$scope.ChangeCounter = function(flag){

if(flag)

$scope.counter++;

else

$scope.counter--;

};

$scope.ChangeStatus = function(status){

if(status)

$scope.statusText="Checked";

else

$scope.statusText="Unchecked";

};

});

</script>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AngJsDemoFormValidation\_003.html\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<html ng-app="formApplication">

<head>

<title>Form Validationdemo</title>

<style>

.error{

color:red;

font-weight:bold;

}

</style>

</head>

<body bgcolor="pink">

<div data-ng-controller="FormController">

<h3>Enter UR Details</h3>

<form name="frm" ng-submit="submitForm()">

<table>

<tr>

<td>Enter Id:</td>

<td> <input type="text" name="txtId" ng-model="person.id"

ng-minlength="5" ng-maxlength="8" ng-required="true"/></td>

<td>

<span class="error"

ng-show="frm.txtId.$dirty && frm.txtId.$error.required">

Mandator Id

</span>

<span class="error"

ng-show="frm.txtId.$dirty && frm.txtId.$error.minlength">

Min 5 digits re

</span>

<span class="error"

ng-show="frm.txtId.$dirty && frm.txtId.$error.maxlength">

Maximum 8 digits Re </span>

</td>

</tr>

<tr>

<td> Enter First Name</td>

<td><input type="text" name="txtFName" ng-model="person.fname"

ng-pattern="/^[a-zA-Z]/" ng-required="true">

</td>

<td>

<span class="error"

ng-show="frm.txtFName.$dirty && frm.txtFName.$error.pattern">

Only Charcteres are allowed

</span>

<span class="error"

ng-show="frm.txtFName.$dirty && frm.txtFName.$error.required">

Mandatory Field </span>

</td>

</tr>

<tr>

<input type="submit" ng-disabled="frm.$invalid" value="Submit Data"

></input>

</tr>

</table>

</form>

</div>

</body>

<script src="Scripts/angular.js"></script>

<script src="Scripts/angular-sanitize.js"></script>

<script type="text/javascript">

var formApp=angular.module("formApplication",[]);

formApp.controller("FormController",function($scope){

$scope.person={'id':0,fname:''};

var result="Data Is entered successfully\n"

$scope.submitForm=function(){

result+="Id Is : "+$scope.person.id;

result+="Name Is : "+$scope.person.fname;

alert(result);

} ;

}

);

</script>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AngularJsDemoCustDirecGreet\_004.html\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<!-- chrome.exe -allow-file-access-from-files -disable-web-security -->

<html ng-app="direApp">

<head>

<style type="text/css">

.customStyle{

color:red;

background-color:yellow;

font-weight:bold;

font-family:candara;

font-size:20pt;

}

</style>

</head>

<body ng-app="direApp" bgcolor="pink">

<hr/>

<h1 ng-init="greeting = 'Hello World'">The greeting is: {{ greeting }}</h1>

<hr/>

<div>

<greet-me>\*\*\*\*\*\*\*\*\*\*\*\*\*\*</greet-me>

</div>

<hr/>

<div greet-me>\*\*\*\*\*\*\*\*\*\*\*\*\*\* </div>

<hr/><hr/>

<div data-greet-me>\*\*\*\*\*\*\*\*\*\*\*\*\*\* </div>

<hr/><hr/>

<div class="greet-me">\*\*\*\*\*\*\*\*\*\*\*\*\*\*</div>

<hr/><hr/>

<div ng-controller="DirectiveController">

<input type="button" greet-me="sayHi()" value=" Call Controller sayHi Function from drective"></input>

</div>

<hr/><hr/>

<div ng-controller="DirectiveController">

<input type="button" change-name value="Change Your Name"/>

<h1>Updated Name : {{empName}}</h1>

</div>

<hr/><hr/>

<div my-anchor></div>

<hr/><hr/>

<div my-next-anchor my-url="Welcome.html" my-link-text="Click me to go to Same Welcome"> </div>

<div>{{myUrl}}</div>

<a href="{{myUrl}}">Again go to welcome page</a>

<hr/><hr/>

<div>

<input type="button" mouseentry mouseleft datastyle="customStyle"

value="MouseEnter / MouseLeave"/>

</div>

<hr/><hr/>

<hr/>

</body>

<script src="Scripts/angular.js"></script>

<script type="text/javascript">

var app = angular.module("direApp",[]);

/\*--------------------------Controller------------------------------\*/

app.controller("DirectiveController",function($scope){

$scope.empName="Vaishali";

$scope.myUrl = "";

$scope.myLinkText = "";

$scope.sayHi=function(){

alert(" Hi :"+$scope.empName);

} ;

$scope.sayBye=function(){

alert(" Bye :"+$scope.empName);

} ;

}

);

/\*--------------------------E------------------------------\*/

app.directive("greetMe",function(){

return {

restrict:'E',

template: '<h1>Welcome To IGate Globle Sol Ltd. Using Element</h1>'

}

});

/\*--------------------------A------------------------------\*/

app.directive("greetMe",function(){

return {

restrict:"A",/\*Default\*/

template: "<h1>Welcome To IGate Globle Sol Ltd.Using Attribute</h1>"

}

});

/\*--------------------------C------------------------------\*/

app.directive("greetMe",function(){

return {

restrict:"C",/\*Default\*/

template: "<h1>Welcome To IGate Globle Sol Ltd.Using Class</h1>"

}

});

/\*--------------------------C------------------------------\*/

app.directive("greetMe",function(){

return function(scope,element,attrs){

element.bind('click',function(){

scope.$apply(attrs.greetMe);

}

)

}

}

);

/\*--------------------------------------------------------\*/

app.directive('changeName',function(){

return function(scope,element,attrs){

/\*Setting Watch Explicitly\*/

scope.$watch('empName', function(newValue, oldValue) {

console.log("Old Value : "+ oldValue+" New Value : "+newValue);

});

element.bind('click',function(){

scope.empName = "Srivastava";

scope.$apply(); //Triggering Digest Cycle

});

}

});

/\*----------------------------------------------------------------\*/

app.directive('myAnchor', function() {

return {

restrict: 'A',

replace: true,

template: '<a href="Welcome.html">Click me to go to Welcome Page</a>'

}

});

/\*----------------------directive as expression--------------------------------\*/

app.directive('myNextAnchor', function() {

return {

restrict: 'A',

template: '<a href="{{myUrl}}">{{myLinkText}}</a>',

link:function(scope,element,attrs){

scope.myUrl = attrs.myUrl;

scope.myLinkText = attrs.myLinkText;

}

}

});

**/\*---------------------MouseEntry---------------------------------\*/**

app.directive("mouseentry",function(){

return {

link:function(scope,element,attrs){

element.bind('mouseenter',function(){

element.addClass(attrs.datastyle);

})

}

}

});

**/\*---------------------MouseMove---------------------------------\*/**

app.directive("mouseleft",function(){

return function(scope,element,attrs){

element.bind('mouseleave',function(){

element.removeClass(attrs.datastyle);

})

}

});

</script>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AngJsDemoCompileService\_005.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<!-- The compile() function sets a border on the HTML element. This is only executed once because the compile() function is only executed once.

The link() function replaces the content of the HTML element, and sets the background color to yellow.

There is no particular reason why we need to set the border in the compile() function,

and the background color in the link() function.

Both could have been set in the compile() function, or both in the link() function.

If set in the compile() function they would only have been set once. If set in the link() function they would be set every time the HTML element is bound to data in the $scope object. This might be useful if you needed to set the border and background color differently depending on data in the $scope object. -->

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd">

<html ng-app="myapp" >

<head>

<meta http-equiv="Content-Type" content="text/html; charset=ISO-8859-1">

<title>Compile Service Demo</title>

</head>

<body bgcolor="pink">

<div ng-controller="MyController" >

<userinfo >This will be replaced</userinfo>

</div>

<userinfo >This will be replaced</userinfo>

<hr/>

</body>

<script type="text/javascript" src="Scripts/angular.js"></script>

<script type="text/javascript" >

var myapp = angular.module("myapp", []);

myapp.directive('userinfo', function() {

var directive = {};

directive.restrict = 'E'; /\* restrict this directive to elements \*/

directive.compile = function(element, attributes) {

element.css("border", "1px solid #cccccc");

var linkFunction = function($scope, element, attributes) {

element.html("This is the new content: " + $scope.firstName);

element.css("background-color", "#ffff00");

}

return linkFunction;

}

return directive;

});

myapp.controller("MyController", function($scope, $http) {

$scope.firstName = "Vaishali";

});

</script>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AngularJsDemoParentChildContrScop\_011.java\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<!-- chrome.exe -allow-file-access-from-files -disable-web-security -->

<html ng-app="myApp">

<head>

</head>

<body bgcolor="pink">

<div ng-controller="EmpController">

{{ empName }}

<button ng-click="setEmpName()">Set Employee Name</button>

<div ng-controller="ManagerController">

{{ empName }}

<button ng-click="setManagerName()">Set Manager Name</button>

</div>

</div>

<hr/>

</hr/>

<div ng-controller="EmpController">

{{ empName }}

<button ng-click="setEmpName()">Set Employee Name</button>

</div>

<div ng-controller="ManagerController">

{{ empName }}

<button ng-click="setManagerName()">Set Manager Name</button>

</div>

<h3>\*\*\*\*\*\*\*\*\*\*\*\*\*\*With CarController\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*</h3>

<div ng-init="carName= 'Honda City'"></div>

<div ng-init="carMialage = '1100CC'">

1 Car Average: {{ carAverage }}

<div ng-init="carAverage = '20Km/Hr'" ng-controller="CarController">

2 Car Average: {{ carAverage }}

<div ng-init="isLuxuryCar">

3 Car Average:: {{ carAverage }}

</div>

</div>

</div>

<h3>\*\*\*\*\*\*\*\*\*\*\*\*\*\*With NextCarController\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*</h3>

<div ng-init="carName= 'Honda City'"></div>

<div ng-init="carMialage = '1100CC'">

1 Car Average: {{ carAverage }}

<div ng-init="carAverage = '20Km/Hr'" ng-controller="CarController">

2 Car Average: {{ carAverage }}

<div ng-init="isLuxuryCar" ng-controller="NextCarController">

3 Car Average:: {{ carAverage }}

</div>

</div>

</div>

</body>

<script src="Scripts/angular.js"></script>

<script type="text/javascript">

angular.module('myApp', [])

.controller('EmpController', function($scope) {

$scope.empName = 'UnKnown';

$scope.setEmpName = function() {

$scope.empName = 'Emp Name Is Vaishali';

};

})

.controller('ManagerController', function($scope) {

$scope.setManagerName = function() {

// sets {{ someBareValue }} inside ChildController

$scope.empName = 'Manager name Is Samir';

};

})

.controller('CarController', function($scope) {

})

.controller('NextCarController', function($scope) {

});

</script>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AngularJsDemoParentChildSiblingDirecScope\_012.java\*\*\*\*\*\***

<!-- chrome.exe -allow-file-access-from-files -disable-web-security -->

<html ng-app="myApp">

<head>

</head>

<body bgcolor="pink">

<div ng-controller="EmpController">

{{ empName }}

<button ng-click="setEmpName()">Set Employee Name</button>

<div ng-controller="ManagerController">

{{ empName }}

<button ng-click="setManagerName()">Set Manager Name</button>

</div>

</div>

<hr/>

</hr/>

<div ng-controller="EmpController">

{{ empName }}

<button ng-click="setEmpName()">Set Employee Name</button>

</div>

<div ng-controller="ManagerController">

{{ empName }}

<button ng-click="setManagerName()">Set Manager Name</button>

</div>

<h3>\*\*\*\*\*\*\*\*\*\*\*\*\*\*With CarController\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*</h3>

<div ng-init="carName= 'Honda City'"></div>

<div ng-init="carMialage = '1100CC'">

1 Car Average: {{ carAverage }}

<div ng-init="carAverage = '20Km/Hr'" ng-controller="CarController">

2 Car Average: {{ carAverage }}

<div ng-init="isLuxuryCar">

3 Car Average:: {{ carAverage }}

</div>

</div>

</div>

<h3>\*\*\*\*\*\*\*\*\*\*\*\*\*\*With NextCarController\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*</h3>

<div ng-init="carName= 'Honda City'"></div>

<div ng-init="carMialage = '1100CC'">

1 Without Controller Car Average : {{ carAverage }}

<div ng-init="carAverage = '20Km/Hr'" ng-controller="CarController">

2 CarController Car Average : {{ carAverage }}

<div ng-init="isLuxuryCar" ng-controller="NextCarController">

3 In NextController Car Average: {{ carAverage }}

</div>

</div>

</div>

<h3>With Directive\*\*\*\*\*\*\*\*\*\*\*\*\*\*scope is optional.

To set properties on the inner scope of our directive, we have a few options. The simplest option is to simply use the existing scope currently provided by the controller (ng-controller) inside of which we’re nested.

While simple, however, sharing state leaves us vulnerable. If that controller is removed or if a

property with the name myUrl is later defined on its scope, we’ll be forced to change our code,

which is costly and frustrating.

To overcome this common issue, Angular provides the ability to create a new child scope or create an isolate scope. In contrast to It can be set to true or to an object, {}. By default, it is set to false.When scope is set to true, a new scope object

is created that prototypically inherits from its parent scope.

When we set the scope of our directive to a clean object with its own properties,

Introduction to Directives 72

scope: {

someProperty: "needs to be set"

}

We’re creating what is referred to as an isolate scope. Essentially, that means the directive gets

its own $scope object, which we can only use inside other methods of the directive or inside the

directive’s template string:

If multiple</h3>

<div ng-init="carName= 'Honda City'"></div>

<div ng-init="carMialage = '1100CC'">

1 Without Controller Car Average : {{ carAverage }}

<div ng-init="carAverage = '20Km/Hr'" ng-controller="CarController">

2 CarController Car Average : {{ carAverage }}

<div ng-init="isLuxuryCar" ng-controller="NextCarController">

3 In NextController Car Average: {{ carAverage }}<br>

4 Outside Directive Car Color Is : {{ carColor }}

<div set-colorr ng-init="carColor = 'wow, Red is cool'">

5 Inside Directive Car Color Is : {{ carColor }}

</div>

</div>

</div>

</div>

<hr/>

<div ng-init="carName= 'Honda City'"></div>

<div ng-init="carMialage = '1100CC'">

1 Without Controller Car Average : {{ carAverage }}

<div ng-init="carAverage = '20Km/Hr'" ng-controller="CarController">

2 CarController Car Average : {{ carAverage }}

<div ng-init="isLuxuryCar" ng-controller="NextCarController">

3 In NextController Car Average: {{ carAverage }}<br>

4 Outside Directive Car Color Is : {{ carColor }}

<div set-colorb ng-init="carColor = 'wow, Blue is cool'">

5 Inside Directive Car Color Is : {{ carColor }}

</div>

</div>

</div>

</div>

<hr/>

<div ng-init="carName= 'Honda City'"></div>

<div ng-init="carMialage = '1100CC'">

1 Without Controller Car Average : {{ carAverage }}

<div ng-init="carAverage = '20Km/Hr'" ng-controller="CarController">

2 CarController Car Average : {{ carAverage }}

<div ng-init="isLuxuryCar" ng-controller="NextCarController">

3 In NextController Car Average: {{ carAverage }}<br>

4 Outside Directive Car Color Is : {{ carColor }}

<div set-colorg ng-init="carColor = 'wow, Green is cool'">

</div> </div> </div>

</div>

<hr/>

</body>

<script src="Scripts/angular.js"></script>

<script type="text/javascript">

angular.module('myApp', [])

.controller('EmpController', function($scope) {

$scope.empName = 'UnKnown';

$scope.setEmpName = function() {

$scope.empName = 'Emp Name Is Vaishali';

};

})

.controller('ManagerController', function($scope) {

$scope.setManagerName = function() {

// sets {{ someBareValue }} inside ChildController

$scope.empName = 'Manager name Is Samir';

}; })

.controller('CarController', function($scope) {

})

.controller('NextCarController', function($scope) {

})

.directive('setColorr', function() {

return {

restrict: 'A',

scope: false,

template: '<div>This is Set Through directive {{ carColor }}</div>'

}

})

.directive('setColorb', function() {

return {

restrict: 'A',

scope: true,

template: '<div>This is Set Through directive {{ carColor }}</div>'

}

})

.directive('setColorg', function() {

return {

restrict: 'A',

scope: {},

template: '<div>This is Set Through directive {{ carColor }}</div>'

}

})

</script>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AngularJsDemoCustDirecIsoscaleSco\_014.html\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<html ng-app="myApp">

<head>

<title>isolated scope - one way binding using '@'</title>

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />

</head>

<body>

<div ng-controller="myController">Controller Says name is :{{name}}

<data-show-name data-getname="{{name}}"></data-show-name>

<data-print-name data-getname="{{name}}"></data-print-name>

<data-display-name data-getname="{{name}}"></data-display-name>

Controller Says name is :{{name}}

</div>

</body>

<script type="text/javascript" src="Scripts/angular.js"></script>

<script>

var app= angular.module("myApp",[]);

app.controller("myController",function($scope){

$scope.name="controller Karthik M";

})

app.directive("showName",function(){

return{

restrict: 'E',

template: "<h1>{{name}}</h1>" ,

replace : true,

scope : true,

link : function(scope,element,attrs){

//scope.name=attrs.getname;

scope.name="showName scope true Vaishali";

}

}

})

app.directive("printName",function(){

return{

restrict: 'E',

template: "<h1>{{name}}</h1>" ,

replace : true,

scope : {},

link : function(scope,element,attrs){

//scope.name=attrs.getname;

scope.name="print Name{} Srivastava";

}

}

})

app.directive("displayName",function(){

return{

restrict: 'E',

replace : true,

scope : '@' ,//one way binding

template: "<h3>dispalyName @{{name}}</h3>"

}

})

</script>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AngularJsDemoCustDirecIsoscaleSco\_015.html\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<html ng-app="directivesApp">

<head>

<title>Custom Directives - usage of scope '=' symbol</title>

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />

</head>

<body>

<h1>Custom Directives Isoscale Scope</h1>

<hr/>

<div ng-controller="AccCntrl">

Bank Admin Says Account Type Is :

<input type="text" ng-model="accounttype"/>

Customer Says My Account Type Is :

<input type="text" account acctype="accounttype"/>

</div>

</body>

<script type="text/javascript" src="Scripts/angular.js"></script>

<script type="text/javascript" >

var app = angular.module("directivesApp",[]);

app.controller('AccCntrl',function($scope){

$scope.accounttype = "Savings Account";

})

app.directive('account',function(){

return{

replace:true,

scope:{

acctype:"=" //two binding

},

template: '<input type="text" ng-model="acctype">'

}

})

</script>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AngularJsDemoCustDirecIsoscaleSco\_016.html\*\*\*\*\*\*\*\*\*\*\*\***

<html ng-app="myApp">

<head>

<title>isolated scope - Two way binding using '='</title>

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />

</head>

<body>

<div ng-controller="myController">

Admin Says Customer Account Type Is :

<input type="text" ng-model="accounttype"/><br/>

Customer Says My Account Type Is :

<data-account data-getaccounttype="accounttype"></data-data-account>

<span>{{accounttype}}</span>

</div>

</body>

<script type="text/javascript" src="Scripts/angular.js"></script>

<script>

var app= angular.module("myApp",[]);

app.controller("myController",function($scope){

$scope.accounttype="Saving Account";

})

app.directive("account",function(){

return{

restrict: 'E',

replace : true,

//scope : { getname: '='} ,

scope : '=',

template: '<input type="text" ng-model="accounttype"/>'

}

})

</script>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AngularJsDemoCustDirecIsoscaleSco\_017.html\*\*\*\*\*\*\*\*\***

<html ng-app="directivesApp">

<head>

<title>Custom Directives</title>

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />

</head>

<body>

<h1>Custom Directives - usage of scope '&amp;' symbol</h1>

<hr/>

<div ng-controller="MobileCntrl">

<div mobile type-Message="sendMessage(msg)"></div>

<div mobile type-Message="sendMessage(msg)"></div>

<div mobile type-Message="sendMessage(msg)"></div>

</div>

</body>

<script type="text/javascript" src="Scripts/angular.js"></script>

<script type="text/javascript" >

var app = angular.module("directivesApp",[]);

app.controller('MobileCntrl',function($scope){

$scope.sendMessage = function(msg){

alert('Recieved : '+msg);

}

})

app.directive('mobile',function(){

return{

scope:{

typeMessage:"&" /\*Invoke Expression. Need to call it as type-Message in view\*/

},

template: '<input type="text" ng-model="message">'+

'<input type="button" ng-click="typeMessage({msg:message})" value="Send Message" />'

}

})

</script>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AngularJsDemoCustDirecIsoscaleSco\_018.html\*\*\*\*\*\*\*\*\*\*\*\*\***

<html ng-app="myApp">

<head>

<title>isolated scope - Execute function in the parent scope using '&'</title>

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />

</head>

<body>

<div ng-controller="myController">

<input type="button" value="IGate Gives Instruction" ng-click="fillFeedback()"/>

<div data-igate-task data-invoke="fillFeedback()"></div>

</div>

</body>

<script type="text/javascript" src="Scripts/angular.js"></script>

<script>

var app= angular.module("myApp",[]);

app.controller("myController",function($scope){

$scope.fillFeedback=function(){

alert('Please Fill FeedBack Form');

}

})

app.directive("igateTask",function(){

return {

template : '<input type="button" value="IGate University Gives Instruction" ng-click="invoke()"/>',

scope : { invoke: '&'}

}

})

</script>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AngularJsDemoCustDirecIsoscaleSco\_019.html\*\*\*\*\*\*\*\*\*\*\*\***

<html ng-app="myApp">

<head>

<title>Custom Directives</title>

</head>

<body>

<h1>Custom Directives</h1>

<hr/>

<div ng-controller="MyCntrl">

<div my-directive var="{{company}}"></div>

</div>

</body>

<script type="text/javascript" src="Scripts/angular.js"></script>

<script>

var app = angular.module("myApp",[]);

app.directive("myDirective", function(){

return {

restrict:"A",

scope: {

company:"@var"

},

template: '{{company}}'

}

});

app.controller("MyCntrl",function($scope){

$scope.company = "IGATE";

});

</script>

</html>

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*AngularJsDemoCustDirecIsoscaleScoTranclusi\_020.html\*\*\*\*\*\*\*\*\*\*\*\***

transclude is optional. If provided, it must be set to true. It is set to false by default.

Transclusion is sometimes considered an advanced topic, but once it makes sense it’ll fit right in

with the rules we’ve just learned about scope. We’ll also see that it’s a very powerful addition to our

tool set, especially when building customizable chunks of HTML that can be shared with a team,

between projects, and with the rest of the Angular community.

Transclusion is most often used for creating reusable widgets. A great example is a modal box or a

navbar.

Transclude allows us to pass in an entire template, including its scope, to a directive. Doing so gives

us the opportunity to pass in arbitrary content and arbitrary scope to a directive. In order for scope

⁵⁸http://jsbin.com/OxAlek/1/edit

Directives Explained 116

to be passed in, the scope option must be isolated, {}, or set to true. If the scope option is not set,

then the scope available inside the directive will be applied to the template passed in.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*&&&&&&&&\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

<html ng-app="directivesApp">

<head>

<title>Custom Directives</title>

<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1" />

<style type="text/css">

.style01{

background-color:crimson;

padding:10px;

text-align:center;

color:white;

}

.style02{

background-color:blue;

padding:14px;

text-align:left;

color:white;

}

</style>

</head>

<body>

<h1>Custom Directives - transclude</h1>

<hr/>

<div>

<div wrapdiv customstyle="style01">

<input type="button" value="Wrap this button with custom style2"/>

</div>

</div>

<hr/>

<div>

<div wrapdiv customstyle="style02">

<input type="button" value="Wrap this button with custom style2"/>

</div>

</div>

</body>

<script type="text/javascript" src="Scripts/angular.js"></script>

<script type="text/javascript" ">

var app = angular.module("directivesApp",[]);

app.directive('wrapdiv',function(){

return{

scope:{

customstyle:"@"

},

transclude:true,

template:'<div class="{{customstyle}}"><h1>WrapDiv Directive</h1><div ng-transclude></div></div>'

}

})

</script>

</html>

****