# Argularjs

## 定义

AngularJS 通过 **ng-directives** 扩展了 HTML。

**ng-app** 指令定义一个 AngularJS 应用程序。

**ng-model** 指令把元素值（比如输入域的值）绑定到应用程序。

ng-model 指令用于绑定应用程序数据到 HTML 控制器(input, select, textarea)的值。

**ng-model** 指令也可以：

* 为应用程序数据提供类型验证（number、email、required）。
* 为应用程序数据提供状态（invalid、dirty、touched、error）。
* 为 HTML 元素提供 CSS 类。
* 绑定 HTML 元素到 HTML 表单。

**ng-bind 指令把应用程序数据绑定到 HTML 视图。**

AngularJS 表达式写在双大括号内：**{{ expression }}**。

AngularJS 表达式把数据绑定到 HTML，这与 **ng-bind** 指令有异曲同工之妙。

AngularJS 将在表达式书写的位置"输出"数据。

## 框架案例

0//引入js文件

<script src="http://cdn.static.runoob.com/libs/angular.js/1.4.6/angular.min.js"></script>

1//定义angularjs模块Module

html:

<div ng-app="myapp">

</div>

js:

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

2//定义angularjs控制器Controller

html:

<div ng-app="myapp" ng-controller="mycontroller">

<input type="text" ng-model="ng-model-value" >

您输入的值是：{{ng-model-value}}//双向绑定

您输入的值是：<h1>{{ng-model-value}}</h1>//双向绑定

</div>

//js:**$scope** 对象当作一个参数传递

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

$scope.ng-model-value = '123';

});

//绑定数据

//初始化数据的方式。

<div ng-app="" ng-init="quantity='2';cost=5,person={firstName:'John',lastName:'Doe'}，points=[1,15,19,2,40]">

<p>总价: <span ng-bind="quantity \* cost"></span> </p>

<p>总价: <span > {{" " +quantity+cost}} </span> </p>

<p>姓为 <span ng-bind="person.lastName"></span></p>

<p>第三个值为 {{ points[2] }}</p>

</div>

输出：10 25

## 循环html

<div data-ng-app="" data-ng-init="names=['Jani','Hege','Kai','aa']">

<p>使用 ng-repeat 来循环数组</p>

<ul>

<li data-ng-repeat="x in names">

{{ x }}

</li>

</ul>

</div>

对象数组

<div ng-app="" ng-init="names=[  
{name:'Jani',country:'Norway'},  
{name:'Hege',country:'Sweden'},  
{name:'Kai',country:'Denmark'}]">  
<p>循环对象：</p>  
<ul>  
  <li ng-repeat="x in names">  
    {{ x.name + ', ' + x.country }}  
  </li>  
</ul>  
</div>

## 自定义指令

使用 **.directive** 函数来添加自定义的指令。

你可以通过以下方式来调用指令：

* 元素名
* 属性
* 类名
* 注释

<body ng-app="myapp">

//元素名

<runoob-directive></runoob-directive>

//属性

<div runoob-directive></div>

//类名

<div class="runoob-directive"></div>

//注释

<!-- directive: runoob-directive -->

<runoob-directive2></runoob-directive2>

<script>

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

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

return {template:"<h1 style='color:#0f0;'>自定义指令</h1>"};

});

app.directive("runoobDirective2",function(){

return {template:"<h1>自定义指令</h1>"};

});

</script>

</body>

**restrict** 值可以是以下几种:

* E 作为元素名使用
* A 作为属性使用
* C 作为类名使用
* M 作为注释使用

在返回函数中

return {  
        restrict : "A",  
        template : "<h1>自定义指令!</h1>"  
 };

## 验证

必须在form表单里验证

Email:

<form ng-app="" name="myForm">

Email:

<input type="email" name="myAddress" ng-model="text">

<span ng-show="myForm.myAddress.$error.email">不是一个合法的邮箱地址</span>

</form>

## 应用状态

<form ng-app="" name="myForm" ng-init="myText = 'test@runoob.com'">

Email:

<input type="email" name="myAddress" ng-model="myText" required>

<p>编辑邮箱地址，查看状态的改变。</p>

<h1>状态</h1>

<p>Valid: {{myForm.myAddress.$valid}} (如果输入的值是合法的则为 true)。</p>

<p>Dirty: {{myForm.myAddress.$dirty}} (如果值改变则为 true)。</p>

<p>Touched: {{myForm.myAddress.$touched}} (如果通过触屏点击则为 true)。</p>

</form>

## **$scope**

//$scope的数组属性

<div ng-app="myApp" ng-controller="myCtrl">

<ul>

<li ng-repeat="x in names">{{x}}</li>

</ul>

</div>

<script>

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

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

$scope.names = ["Emil", "Tobias", "Linus"];

});

</script>

## $rootScope

<div ng-app="myApp" ng-controller="myCtrl">

<ul>

<li ng-repeat="x in names">{{x}} {{lastname}}</li>

</ul>

<h1>姓氏为 {{lastname}} 家族成员:</h1>

</div>

<script>

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

app.controller('myCtrl', function($scope, $rootScope) {

$scope.names = ["Emil", "Tobias", "Linus"];

$rootScope.lastname = "Refsnes";

});

</script>

<p>注意 $rootScope 在循环对象内外都可以访问。</p>

## ng-app

AngularJS 应用程序由 **ng-app** 定义。应用程序在 <div> 内运行。

**ng-controller="myCtrl"** 属性是一个 AngularJS 指令。用于定义一个控制器。

**myCtrl** 函数是一个 JavaScript 函数。

AngularJS 使用**$scope** 对象来调用控制器。

## 控制器方法

<div ng-app="myApp" ng-controller="personCtrl">

名: <input type="text" ng-model="firstName"><br>

姓: <input type="text" ng-model="lastName"><br>

<br>

姓名: {{fullName()}}

</div>

<script>

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

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

$scope.firstName = "John";

$scope.lastName = "Doe";

$scope.fullName = function() {

return $scope.firstName + " " + $scope.lastName;

}

});

</script>

## 表达式中添加过滤器

<p>姓名为 {{ lastName | uppercase }}</p>

## $location 服务

var app = angular.module('myApp', []);  
app.controller('customersCtrl', function($scope, $location) {  
    $scope.myUrl = $location.absUrl();  
});

## $http 服务

//$http.get(url) 是用于读取服务器数据的函数

<div ng-app="myApp" ng-controller="myCtrl">

<h1>{{myWelcome}}</h1>

</div>

<p> $http 服务向服务器请求信息，返回的值放入变量 "myWelcome" 中。</p>

<script>

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

app.controller('myCtrl', function($scope, $http) {

$http.get("welcome.htm").then(function (response) {

$scope.myWelcome = response.data;

});

});

</script>

//读取服务器的json数据

<div ng-app="myApp" ng-controller="siteCtrl">

<ul>

<li ng-repeat="x in names">

{{ x.Name + ', ' + x.Country }}

</li>

</ul>

</div>

<script>

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

app.controller('siteCtrl', function($scope, $http) {

$http.get("http://www.runoob.com/try/angularjs/data/sites.php")

.success(function (response) {

//当从服务端载入 JSON 数据时，**$scope.names** 变为一个数组。

$scope.names = response.sites;

});

});

</script>

## $timeout 服务

<script>

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

app.controller('myCtrl', function($scope, $timeout) {

$scope.myHeader = "Hello World!";

$timeout(function () {

$scope.myHeader = "How are you today?";

}, 2000);

});

</script>

## $interval 服务

<p>$interval 访问在指定的周期(以毫秒计)来调用函数或计算表达式。</p>

<script>

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

app.controller('myCtrl', function($scope, $interval) {

$scope.theTime = new Date().toLocaleTimeString();

$interval(function () {

$scope.theTime = new Date().toLocaleTimeString();

}, 1000);

});

</script>

## 创建自定义服务

//'hexafy'服务名

app.service('hexafy', function() {  
    this.myFunc = function (x) {  
        return x.toString();  
    }  
});

## 在控制器中使用使用服务

app.controller('myCtrl', function($scope, **hexafy**) {  
    $scope.hex = **hexafy**.myFunc(255);  
});

## 在过滤器中使用服务

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

app.service('hexafy', function() {

this.myFunc = function (x) {

return x.toString(16);

}

});

app.filter('myFormat',['hexafy', function(hexafy1) {

return function(x) {

return hexafy1.myFunc(x);

};

}]);

## Select(选择框)

使用数组或对象创建一个下拉列表选项。

## 使用 ng-options 创建选择框

<div ng-app="myApp" ng-controller="myCtrl">

<select ng-model="selectedName" ng-options="x for x in names">

</select>

</div>

<script>

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

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

$scope.names = ["Google", "Runoob", "Taobao"];

$scope.selectedName = $scope.names[0];

});

</script>

<select ng-model="selectedSite" ng-options="x.site for x in sites">

</select>

<h1>你选择的是: {{selectedSite.site}}</h1>

<p>网址为: {{selectedSite.url}}</p>

</div>

<script>

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

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

$scope.sites = [

{site : "Google", url : "http://www.google.com"},

{site : "Runoob", url : "http://www.runoob.com"},

{site : "Taobao", url : "http://www.taobao.com"}

];

});

</script>

//对象

<h1>你选择的是: {{selectedCar.brand}}</h1>

<h2>模型: {{selectedCar.model}}</h2>

<h3>颜色: {{selectedCar.color}}</h3>

<p>注意选中的值是一个对象。</p>

</div>

<script>

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

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

$scope.cars = {

car01 : {brand : "Ford", model : "Mustang", color : "red"},

car02 : {brand : "Fiat", model : "500", color : "white"},

car03 : {brand : "Volvo", model : "XC90", color : "black"}

}

});

</script>

<select ng-model="selectedCar" ng-options="y.brand for (x, y) in cars"></select>

<p>你选择的是: {{selectedCar.brand}}</p>

<p>型号为: {{selectedCar.model}}</p>

<p>颜色为: {{selectedCar.color}}</p>

<p>下拉列表中的选项也可以是对象的属性。</p>

</div>

<script>

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

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

$scope.cars = {

car01 : {brand : "Ford", model : "Mustang", color : "red"},

car02 : {brand : "Fiat", model : "500", color : "white"},

car03 : {brand : "Volvo", model : "XC90", color : "black"}

}

});

</script>

## 使用ng-repeat 指令来创建下拉列表

<div ng-app="myApp" ng-controller="myCtrl">

<select>

<option ng-repeat="x in names">{{x}}</option>

</select>

</div>

<script>

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

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

$scope.names = ["Google", "Runoob", "Taobao"];

});

</script>

## 表格

<div ng-app="myApp" ng-controller="customersCtrl">

<table>

<tr ng-repeat="x in names">

<td>{{ x.Name }}</td>

<td>{{ x.Country }}</td>

</tr>

</table>

</div>

<script>

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

app.controller('customersCtrl', function($scope, $http) {

$http.get("/try/angularjs/data/Customers\_JSON.php")

.success(function (response) {$scope.names = response.records;});

});

</script>

## 使用 CSS 样式

<style>  
table, th , td {  
  border: 1px solid grey;  
  border-collapse: collapse;  
  padding: 5px;  
}  
table tr:nth-child(odd) {  
  background-color: #f1f1f1;  
}  
table tr:nth-child(even) {  
  background-color: #ffffff;  
}  
</style>

## 显示序号 ($index)

 <td>{{ $index + 1 }}</td>

## ng-disabled 指令

**ng-disabled** 指令直接绑定应用程序数据到 HTML 的 disabled 属性

<div ng-app="" ng-init="mySwitch=false">

<p>

<input type="checkbox" ng-model="mySwitch"/>按钮

</p>

{{ mySwitch }}

</div>