

后盾网 人人做后盾

www.houdunwang.com

Angular JS

后盾网 2011-2015

简介

- AngularJS[1] 诞生于2009年，由Misko Hevery 等人创建，后为Google所收购。是一款优秀的前端JS框架，已经被用于Google的多款产品当中。AngularJS有着诸多特性，最为核心的是：MVVM(Model-View-ViewModel缩写)、模块化、自动化双向数据绑定、语义化标签、依赖注入等等。

AngularJS下载地址

<http://angularjs.org>

或者cdn载入

<http://cdn.bootcss.com/angular.js/1.4.6/angular.js>

文档查看

<http://docs.angularjs.cn/api>

什么是 AngularJS?

```
<script src="http://cdn.bootcss.com/angular.js/1.4.6/angular.js"></script>
  <article ng-app="hd">
    <div ng-controller="ctrl">
      {{name}}
    </div>
  </article>
  <script>
    var m = angular.module('hd', []);
    m.controller('ctrl', ['$scope', function ($scope) {
      $scope.name='后盾网';
    }])
  </script>
```

`$scope`是angularJS中的作用域(其实就是存储数据的地方)

简单的使用

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。

AngularJS 表达式 很像 JavaScript 表达式：它们可以包含文字、运算符和变量。

实例 {{ 5 + 5 }}

- <!DOCTYPE html>
- <html lang="en" ng-app> 简便方式ng-app可以不需要给名称
- <head>
- <meta charset="UTF-8">
- <title>表达式</title>
- <script src="http://www.houdunwang.com/angular.min.js"></script>
- </head>
- <body>
- {{1+2}}
- </body>
- </html>

AngularJS 表达式

\$scope 局部作用域

\$rootScope 全部作用域,多个控制器可共享

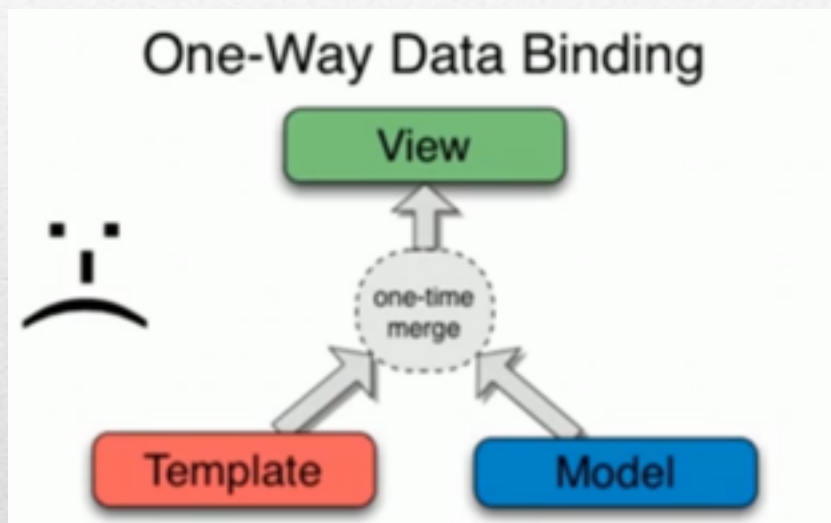
搜索的时候, 优先找自己的scope, 如果没有找到就沿着作用域链向上搜索, 直至到达根作用域rootScope。

示例

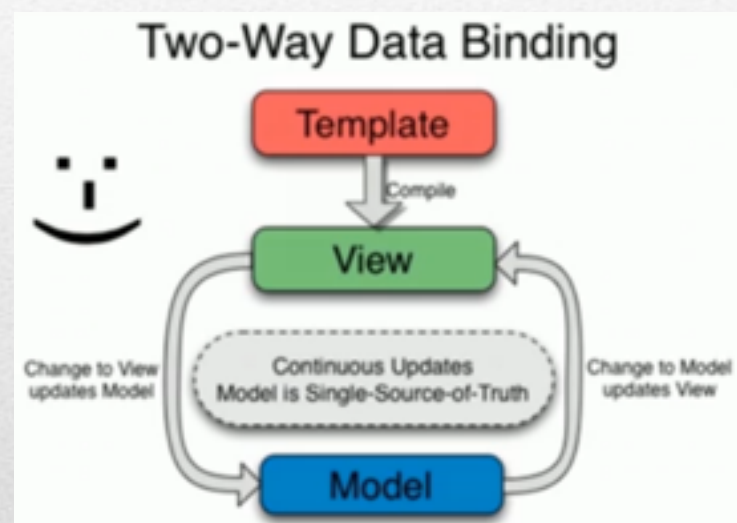
```
<body ng-app="hd">
  <div ng-controller='article'>
    {{name}}
    {{url}}
  </div>
  <p ng-controller="category">
    {{name}}
    {{url}}
  </p>
  <script type="text/javascript">
    var m = angular.module('hd',[]);
    m.controller('article',function($scope,$rootScope){
      $rootScope.name = '我是公用的';
      $scope.url = 'article.com';
    })
    m.controller('category',function($scope,$rootScope){
      $scope.url = 'category.com';
      //alert($rootScope.name);
    })
  </script>
</body>
```

\$scope与\$rootScope

单向数据绑定



双向绑定(angular.js)



双向绑定

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-bind 指令把应用程序数据绑定到 HTML 视图

```
<body ng-app>  
  <input type="text" name="username" id="" ng-model="name"/>  
  <!-- 下面两种用法效果是一样的-->  
  <p ng-bind="name"></p>  
  <p>{{name}}</p>  
</body>
```

数据绑定

ng-init 指令初始化应用程序数据

```
<body ng-app>
  <p ng-init="num1=1;num2=2"></p>
  <!-- 下面两种用法效果是一样的-->
  <p ng-bind="num1+num2"></p>
  <p>{{num1+num2}}</p>
</body>

<body ng-app>
  <p ng-init="a='后盾网';b='php实战培训'"></p>
  <!-- 下面两种用法效果是一样的-->
  <p ng-bind="a+'='>'+b"></p>
  <p>{{a+"=">"+b}}</p>
</body>
```

扩展的HTML属性

```
<!DOCTYPE html>
<html lang="en" ng-app>
<head>
  <meta charset="UTF-8">
  <title>数组</title>
  <script src="http://www.houdunwang.com/angular.min.js"></script>
</head>
<body>
  <input type="text" ng-init="a=['后盾网']">
  <p ng-bind="a[0]"></p>
</body>
</html>
```

扩展的HTML属性

```
<div ng-app="app" ng-controller="hdController">
  <input type="text" ng-model="hd.age">
  <span ng-bind='hd.age'></span>
  <p>{{hd.getName()}}</p>
</div>
<script>
  var app=angular.module('app',[]);
  app.controller('hdController',function($scope){
    $scope.hd = {
      age:'18',
      getName:function()
      {
        return '后盾网';
      }
    };
  });
</script>
```

在大型的应用程序中，通常是把控制器存储在外部文件中。

AngularJS 控制器

angular.run:

angular.run可以看成是angular的构造函数,它能在应用的其他部分之前被执行, 可用于初始化全局域中的数据, 不需要定义控制器即可使用angular.run。

```
<div class="box" ng-app="hd" ng-controller="article">
  <p>{{title}}</p>
</div>
<script type="text/javascript">
  var m = angular.module('hd',[]);
  //执行控制器
  m.controller('article',function($scope,$rootScope){
    //把构造函数的值复制给当前控制器的变量
    $scope.title = $rootScope.title;
  })
  //构造函数, 先执行
  m.run(function($rootScope){//传入$rootScope不能传入$scope
    $rootScope.title = '初始化标题';
  })
</script>
```

构造函数

\$scope.\$watch:

当模型数据发表变化时，触发\$scope.\$watch方法,并传入新值与旧值。

如果数据是一个对象，需要将第三个参数设为true，用于监听对象。

```
<div class="box" ng-app="hd" ng-controller="password">
  <input type="text" name="password" id="" ng-model="pwd" ng-init="pwd=""/>
  <span style="color: red;">{{message}}</span>
</div>

angular.module('hd',[]).controller('password',function($scope){
  $scope.$watch('pwd',function(newP,oldP){
    switch (true){
      case newP.length>10:
        $scope.message = '强! ';
        break;
      case newP.length>5:
        $scope.message = '还行! ';
        break;
      default:
        $scope.message = '给我密码! ';
        break;
    }
  })
})
```

数据变化监听

\$scope.\$apply:

如果数据是延迟设置的，会造成视图中数据不更新，我们可以使用\$scope.\$apply手动进行数据更新。

```
<div id="" ng-app="hd" ng-controller="title">
  <span>{{message}}</span>
</div>
<script type="text/javascript">
  var app = angular.module('hd',[]);
  app.controller('title',function($scope){
    setTimeout(function(){
      $scope.message = '消息';
      //如果没有调取apply那么，数据是2s之后在视图不会更新的
      $scope.$apply();
    },2000);
  })
</script>
```

更新视图数据

之前我们看到使用ng-app指令，可以实现模块的自动加载。如果想更加灵活，那么可以使用手动加载模块。

```
<div id="hd">
  <p ng-controller="hdctrl">{{name}}</p>
</div>
<div id="baidu">
  <p ng-controller="bdctrl">{{name}}</p>
</div>
<script type="text/javascript">
  //定义hd模块
  angular.module('hd', []).controller('hdctrl', function($scope){
    $scope.name = '后盾网';
  })
  //定义baidu模块
  angular.module('baidu', []).controller('bdctrl', function($scope){
    $scope.name = '百度网';
  })
  //手动两个模块（参数1：要定义模块的元素的原生对象，参数2：上面定义的模块的名称）
  angular.bootstrap(document.getElementById("hd"), ['hd']);
  angular.bootstrap(document.getElementById("baidu"), ['baidu']);
</script>
```

手动初始化模块

思考：

自动计算总价的购物车实例。

商品：ip6s

价格：

数量：

商品：ip7s

价格：

数量：

总价：0

思考

```
<div id="price" ng-app>
  商品：ip6s
  <br />
  价格：<input type="text" name="price" id="" ng-model="price1" ng-init="price1=0"/>
  <br />
  数量：<input type="text" name="num" id="" ng-model="num1" ng-init="num1=0"/>
  <br />
  <hr />
  商品：ip6s
  <br />
  价格：<input type="text" name="price" id="" ng-model="price2" ng-init="price2=0"/>
  <br />
  数量：<input type="text" name="num" id="" ng-model="num2" ng-init="num2=0"/>
  <br />
  <hr />
  总价：{{price1 * num1 + price2 * num2}}
</div>
```

计算总价的购物车 实例
