

后盾网 人人做后盾

www.houdunwang.com

Angular JS

后盾网 2011-2016

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、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>

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>
```

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 }} 或 {{ firstName + " " + lastName }}

- <!DOCTYPE html>
- <html lang="en" ng-app>
- <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

示例

```
<div ng-app="hd">
  <div ng-controller="ctrl">
    {{name}}{{name}}
    {{age}}
  </div>
</div>
<script>
  var m = angular.module('hd', []);
  m.controller('ctrl', ['$scope', '$rootScope', function ($scope, $rootScope) {
    $rootScope.url='houdunwang.com';
    $scope.name='后盾网';
  }])
</script>
```

\$scope与\$rootScope服务

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

- `<script src="http://cdn.bootcss.com/angular.js/1.4.6/angular.js"></script>`
- `<div ng-app="hd">`
- `<div ng-controller="ctrl">`
- `<p>在输入框中尝试输入： </p>`
- `<p>姓名： <input type="text" ng-model="name"></p>`
- `<p ng-bind="name"></p>`
- `</div>`
- `</div>`
- `<script>`
- `angular.module('hd', [])`
- `.controller('ctrl', ['$scope', function ($scope) {}])`
- `</script>`

AngularJS扩展了HTML

```
<!DOCTYPE html>
<html lang="en" ng-app>
<head>
  <meta charset="UTF-8">
  <title>数字</title>
  <script src="http://www.houdunwang.com/angular.min.js"></script>
</head>
<body>
  <input type="text" ng-init="a=12;b=33">
    {{a*b}}
  <p ng-bind="a*b"></p>
</body>
</html>
```

AngularJS 数字

```
<!DOCTYPE html>
<html lang="en" ng-app>
<head>
  <meta charset="UTF-8">
  <title>数字</title>
  <script src="http://www.houdunwang.com/angular.min.js"></script>
</head>
<body>
  <input type="text" ng-init="a='后盾网';b='php实战培训'">
  {{a+">" + b}}
  <p ng-bind="a+'>' + b"></p>
</body>
</html>
```

AngularJS 字符串

```
<!DOCTYPE html>
<html lang="en" ng-app>
<head>
  <meta charset="UTF-8">
  <title>数字</title>
  <script src="http://www.houdunwang.com/angular.min.js"></script>
</head>
<body>
  <input type="text" ng-init="a={name:'后盾网',url:'houdunwang.com'}">
  <p ng-bind="a.name+'网址: '+a.url"></p>
</body>
</html>
```

AngularJS 对象

```
<!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>
```

AngularJS 数组

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>控制器</title>
  <script src="http://www.houdunwang.com/angular.min.js"></script>
</head>
<body >
  <div ng-app="app" ng-controller="hdController">
    <input type="text" ng-model="hd.name">
    <span ng-bind='hd.name'></span>
    {{hd.getName()}}
  </div>
  <script>
    var app=angular.module('app',[]);
    app.controller('hdController',function($scope){
      $scope.hd = {
        name:'aaa',
        getName:function()
        {
          return '后盾网';
        }
      };
    });
  </script>
</body>
</html>
```

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。

```
<script src="http://cdn.bootcss.com/angular.js/1.4.6/angular.js"></script>
<div ng-app="hd">
  {{name}}
</div>
<div ng-app="hd" ng-controller="ctrl">
  @{{1+2}}
</div>
<script>
  angular.module('hd', []).controller('ctrl', ['$scope', function($scope){}]);
</script>
```

angular.run() angular的构造函数

\$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。

```
<script src="http://cdn.bootcss.com/angular.js/1.4.6/angular.js"></script>
```

```
<div ng-app="hd">
```

```
  <div ng-controller="ctrl">
```

```
    密码: <input type="text" ng-model="name" ng-init="name=''"><span>{{intensity}}</span>
```

```
  </div>
```

```
</div>
```

```
<script>
```

```
  angular.module('hd', []).controller('ctrl', ['$scope', function ($scope) {
```

```
    $scope.$watch('name', function(newV,oldV){
```

```
      switch(true)
```

```
      {
```

```
        case newV.length>5:
```

```
          $scope.intensity='密码老强了';
```

```
        break;
```

```
        case newV.length>0:
```

```
          $scope.intensity='等着被盗号吧';
```

```
        break;
```

```
        default:
```

```
          $scope.intensity='给个密码呗';
```

```
      }
```

```
    }, false);
```

```
  }]);
```

\$scope.\$watch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。

```
angular.module('hd', []).controller('ctrl', ['$scope', function ($scope) {  
    $scope.name='后盾';  
    setTimeout(function(){  
        //第1种写法  
        // $scope.name='PHP培训';  
        // console.log($scope.name);  
        // $scope.$apply();  
  
        //第2种写法  
        $scope.$apply(function(){  
            $scope.name='PHP培训';  
        })  
    },2000);  
}]);
```

\$scope.\$apply更新视图数据


```
<script src="http://cdn.bootcss.com/jquery/2.1.4/jquery.min.js"></script>
  <script src="http://cdn.bootcss.com/angular.js/1.4.6/angular.js"></script>
  <div id="hd">
    <div ng-controller="ctrl">{{name}}</div>
  </div>
  <div id="baidu">
    <div ng-controller="ctrl">{{name}}</div>
  </div>
</script>
  //第一个模块
  angular.module('hd', []).controller('ctrl', ['$scope', function ($scope) {
    $scope.name='后盾';
  }]);
  //第二个模块
  angular.module('baidu', []).controller('ctrl', ['$scope', function ($scope) {
    $scope.name='百度';
  }])
  //初始各个模块
  angular.bootstrap(angular.element('#hd'), ['hd']);
  angular.bootstrap(angular.element('#baidu'), ['baidu']);
</script>
```

构建多个模块
