

后盾网 人人做后盾

www.houdunwang.com

Angular JS

Directives指令

后盾网 2011-2015

过滤器	描述
ng-app	指令用来初始化AngularJS 应用
ng-controller	实例化控制器
ng-repeat	为对象或数组的每一个元素生成一个对应的HTML元素

基本指令

名称	描述
\$index	从0开始的索引
\$first	是否为第1条数据
\$middle	是否为中间数据
\$last	是否为最后一条数据
\$even	是否为奇数行
\$odd	是否为偶数行

ng-repeat参数说明

ng-repeat 遍历数据:

```
<ul ng-app="hd">
  <li ng-repeat="(k,v) in user | orderBy:id:true">
    {{v.id}}-{{v.name}}-{{v.tel}}
  </li>
</ul>
<script type="text/javascript">
  var app = angular.module('hd',[]);
  //没有控制器只用run, 那么必须使用$rootScope
  app.run(function($rootScope){
    $rootScope.user = [
      {id : 1,tel:15510608116,name:'马震宇'},
      {id : 2,tel:18600276067,name:'向军'}
    ];
  })
</script>
```

ng-repeat与过滤器结合使用

ng-repeat 自定义排序

```
<div id="" ng-app="hd" ng-controller="list">
  <li ng-repeat="(k,v) in user | orderBy:id:true">
    {{v.id}}-{{v.name}}-{{v.tel}}
  </li>
  <button ng-click="sortBy('id')">点我排序</button>
</div>
<script type="text/javascript">
  var app = angular.module('hd',[]);
  app.controller('list',function($scope,$filter){
    $scope.user = [
      {id : 1,name:'马震宇',tel:'15510608116'},
      {id : 2,name:'刘梅老师',tel:'18611400072'}
    ];
    var status = false;
    $scope.sortBy = function(field){
      status = !status;
      //调用排序过滤器
      $scope.user = $filter('orderBy')($scope.user,field,status);
    }
  })
</script>
```

ng-repeat 自定义排序实例

```
<div ng-app="hd">
<input type="text" ng-model="name"/>
<ul style="background: #f7f7f7;">
<li ng-repeat="v in result">{{v.title}}</li>
</ul>
</div>
<script>
var m = angular.module('hd', []);
m.run(function($rootScope,$filter){
    $rootScope.$watch('name',function(newV,oldV){
        //如果为未定义(第一次是未定义)或者为空的时候消失
        if(angular.isUndefined(newV) || newV == ""){
            $rootScope.result = [];
        }else{
            //否则搜索
            $rootScope.result = $filter('filter')($rootScope.data,newV);
        }
    })
    $rootScope.data = [
        {title:'后盾网成功上市了'},{title:'百度2016再创辉煌'},{title:'新浪被马震宇收购了'},{title:'后盾网2016飞黄腾达'},
    ];
})
</script>
```

ng-repeat 结合过滤搜索实例

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

//\$index就是循环时候的索引，可以放到li中间{{\$index}}看结果

```
<li ng-repeat="(key, value) in user track by $index">
    {{value}}
</li>
```

```
</div>
```

```
<script>
```

```
var m = angular.module('hd', []);
m.run(['$rootScope',function ($rootScope) {
    $rootScope.user=[1,2,3,2];
}]);
```

```
</script>
```

ng-repeat不允许存在两个相同id对应的对象

以上代码不使用 track by \$Index 将报错误，因为值有重复的2，track by 后面必须是值唯一的变量

ng-repeat 重复值遍历失败的问题

名称	描述
ng-click	单击时触发
ng-blur	失去焦点时的触发
ng-dbl-click	双击时触发
ng-mousedown	鼠标按下时触发
ng-mouseenter	鼠标进入时触发
ng-mouseleave	鼠标离开元素时触发
ng-mousemove	鼠标移动时触发
ng-mouseup	鼠标抬起时触发
ng-keydown	键盘按下时触发
ng-keyup	键盘抬起时触发
ng-keypress	键盘按下并松开时触发
ng-focus	获取焦点时触发

事件指令

单击同时改变input和span

```
<div id="" ng-app="hd" ng-controller="ctrl">
  <input type="text" name="" id="" ng-model="name"/>
  <span>{{name}}</span>
  <br />
  <button ng-click="name='我有一个梦想'">点我改变input和span</button>
</div>
<script type="text/javascript">
  var app = angular.module('hd',[]);
  app.controller('ctrl',function(){
  })
</script>
```

ng-click 单击事件

失去焦点获得焦点处理方式

```
<div ng-app="hd" ng-controller="ctrl">
  <input type="text" name="msg" id="" ng-model="message" ng-blur="b()" ng-focus="f()"/>
  <span>{{message}}</span>
</div>
<script type="text/javascript">
  var app = angular.module('hd',[]);
  app.controller('ctrl',function($scope){
    $scope.b = function(){
      $scope.message = '别离开我😭';
    }
    $scope.f = function(){
      $scope.message = "";
    }
  })
</script>
```

事件指定实例

名称	描述
ng-selected	表达式为true时选中option
ng-change	数据更新后触发
ng-disabled	表达式为true时表单禁用
ng-readonly	表达式为true时表单只读
ng-checked	表达式为true时选中表单
ng-value	设置表单值,如果使用value="{{name}}"形式网速慢时会显示花扩号
ng-init	令用来初始化application数据
ng-model	将HTML控件(如input,textarea等)的值绑定到angular数据
ng-model-options	配合ng-model使用，指什么时候同步ng-model数据的时机
ng-options	遍历出option标签
ng-trim	将数据左右空白删除

表单指令

ng-selected 选中option:

```
<div ng-app="hd">
  <select>
    //当sex==1的时候被选中
    <option ng-selected="sex==1" value="1">男</option>
    <option ng-selected="sex==2" value="2">女</option>
  </select>
  <input type="radio" ng-model="sex" value="1"> 男
  <input type="radio" ng-model="sex" value="2"> 女
</div>
<script>
  var m = angular.module('hd', []);
  m.run(['$rootScope',function ($rootScope) {
    }]);
</script>
```

ng-selected

ng-init 初始数据:

```
<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" ng-init='name="后盾网"'></p>
    <p ng-bind="name"></p>
  </div>
</div>
<script>
  angular.module('hd', []).controller('ctrl', ['$scope', function ($scope) {}])
</script>
```

ng-init

ng-options 遍历option标签:

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

//s的名字是随便起的, 以as作为中间, 左边为option的value值, 右边为option显示的文本

```
<select name="cname" ng-options="s.value as s.name for s in city" ng-model="selected">
```

```
<option value=""> 请选择城市 </option>
```

```
</select>
```

```
</div>
```

```
<script>
```

```
var m = angular.module('hd', []);
```

```
m.run(['$rootScope', function ($rootScope) {
```

```
    $rootScope.city=[
```

```
        {name:'后盾网',value:'houdunwang',id:1},{name:'北京',value:'beijing',id:2},
```

```
        {name:'上海',value:'shanghai',id:3}
```

```
    ];
```

```
  ]})
```

```
</script>
```

说明:

s.value 值与ng-model绑定, 也就是option的value值

s.name 为显示文本

s 每一个数据对象

ng-options

ng-disabled 设置表单disabled属性：

```
<div ng-app>  
  <input type="checkbox" ng-init="status=false" ng-model="status" > 接受协议  
  <button ng-disabled="!status">注册</button>  
</div>
```

ng-trim 绑定的数据去掉左右空白：

```
<div ng-app>  
  //默认是true，可以改成false观看效果  
  <input type="text" name="mail" ng-trim="true" ng-model="usermail">  
  <pre>{{usermail}}</pre>  
</div>
```

ng-disabled与ng-trim实例

失去焦点时同步ng-model:

```
<div>
  <input ng-model="php" ng-model-options="{updateOn:'blur'}"/>
  {{php}}
</div>
```

ng-model-options 3秒后同步ng-model:

并不想让用户每次按键的时候就立刻更新模型,而是希望当用户输入完了一段有意义的搜索字段以后才更新模型.

```
<input ng-model="php" ng-model-options="{debounce:3000}"/>
{{php}}
```

默认事件基础上添加blur事件, 并设置默认事件更新时间3秒, 失去焦点立刻更新ng-model:

```
//{updateOn:'默认事件 失去焦点事件',debounce:(默认事件:3秒,失去焦点事件:0秒)}
<input ng-model="php" ng-model-options="{updateOn:'default blur',debounce:{default:
  3000,blur:0}}"/>
{{php}}
```

ng-model-options实例

名称	描述
ng-bind	绑定angular数据
ng-bind-html	绑定html代码片段
ng-bind-template	绑定多个数据
ng-cloak	数据绑定前隐藏标签，绑定后显示标签，避免闪烁效应
ng-non-bindable	不解析表达式

绑定数据指令

绑定html标签:

```
<div ng-app="hd">
  <div ng-bind-html="hdHtml"></div>
</div>
<script>
  var m = angular.module('hd', []);
  m.run(['$rootScope','$sce',function ($rootScope,$sce) {
    //使用$sce服务声明html内容为信任
    $rootScope.hdHtml=$sce.trustAsHtml("<span style='color:red'>name</
span>");
  }]);
</script>
```

ng-bind-html实例

ng-bind-template 绑定多个数据:

```
<script src="http://cdn.bootcss.com/angular.js/1.4.6/angular.js"></script>
<div ng-app="hd">
  <pre ng-bind-template="{{url}} {{name}}!"></pre>
</div>
<script>
  var m = angular.module('hd', []);
  m.run(['$rootScope',function ($rootScope) {
    $rootScope.url='houdunwang.com';
    $rootScope.name='后盾网';
  }]);
</script>
```

ng-bind-template实例

在使用表达式{{}}时刷新太快，有时会出现闪烁的问题，angular会在DOM加载完后去解析视图view，在一些快速浏览器中会在angular解析之前dom就已经显示了所以就会出现闪烁的问题

1.使用ng-bind指令

2.使用ng-cloak指令

```
<div ng-app="hd" ng-controller="ctrl">
  <p>
    {{name}}
  </p>
  <p ng-bind="name"></p>
  <p ng-cloak>{{name}}</p>
</div>
```

```
<script type="text/javascript">
```

```
//注释掉这些代码，就可以看到第一种情况显示，其他两种都是隐藏的，避免了闪烁问题
//var app = angular.module('hd',[]);
//app.controller('ctrl',function($scope){
//    $scope.name = '姓名';
//})
```

```
</script>
```

闪烁解决方案

不解析表达式

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

```
<p>使用 AngularJS: {{ 5+5 }}</p>
```

```
<p ng-non-bindable>不使用 AngularJS: {{ 5+5 }}</p>
```

```
</div>
```

不解析表达式

名称	描述
ng-class	绑定angular数据
ng-class-odd	奇数行应用的类
ng-class-even	偶数行应用的类
ng-style	设置行级样式
ng-show	表达式为true时显示元素，即display=block
ng-hide	表达式为true时隐藏元素，即display=none

样式指令

ng-style 设置行样式:

```
<div ng-app>  
  <p ng-style="{color:'red',fontSize:'66px'}">后盾网</p>  
</div>
```

使用style1与style2样式:

```
<style>  
  .style1{color:green;} .style2{font-size:55px;}  
</style>  
<div ng-app ng-class="{style1:true,style2:true}">后盾网 人人做后盾</div>
```

通过模块数据设置:

```
<style>  
  .style1{color:green;} .style2{font-size:55px;}  
</style>  
<div ng-app>  
  应用 style1 类: <input type="checkbox" ng-model="deleted" >  
  应用 style2 类: <input type="checkbox" ng-model="important" >  
  <div ng-class="{style1:deleted,style2:important}">后盾网 人人做后盾</div>  
</div>
```

ng-class使用

奇数行和偶数行应用不同类：

```
<style>
    .style1{color:green;} .style2{color:red;}
</style>
<div ng-app="hd">
    <li ng-repeat="value in data" ng-class-odd=""style1"" ng-class-even=""style2"">
        {{value.name}}
    </li>
</div>
<script>
    var m = angular.module('hd', []);
    m.run(['$rootScope',function ($rootScope) {
        $rootScope.data=[
            {name:'后盾网'}, {name:'百度网'},
            {name:'雅虎网'}, {name:'淘宝网'}
        ];
    }]);
</script>
```

ng-class-odd与ng-class-even实例

ng-show 显示元素：

```
<div ng-app>
```

status不是固定的，可以换成任意值

```
<input type="checkbox" ng-model="status"> 点我显示
```

```
<p ng-show="status">后盾网</p>
```

```
</div>
```

ng-show实例

名称	描述
ng-if	表达式为true时移除dom元素
ng-switch	分支判断指令，类似 js中的switch语句

语句指令

ng-if 添加或移除元素:

```
<div ng-app="">
<input type="checkbox" ng-model="checked" ng-init="checked=true" />
<span ng-if="checked" >
    选中单选框时显示
</span>
</div>
```

ng-switch 分支指令:

```
<div ng-app="hd">
    <input type="radio" ng-model="userType" name="user" value="1"> 企业
    <input type="radio" ng-model="userType" name="user" value="2" ng-checked="true"> 个人
    <div ng-switch on="userType">
        <div ng-switch-default>用户名 <input type="text"></div>
        <div ng-switch-when="1">企业名称 <input type="text"></div>
    </div>
</div>
<script>
    var m = angular.module('hd', []);
</script>
```

ng-if 与 ng-switch实例