

AngularJS 学习心得

AngularJS 指令及自定义指令 directive

ng-app 指令定义了 AngularJS 应用程序的根元素

如图, 在 index.html 中的根元素上设置 ng-app, 在 app.js 中配置整个 app 的基础设置, 包括自定义 module 的注入, 路由的配置等。

```
// Declare app level module which depends on views, and components
angular.module('myApp', ['ngRoute', 'headerModule', 'footerModule', 'mainModule', 'listModule', 'formModule', 'networkModule'])
.config(['$routeProvider', function ($routeProvider) {
  $routeProvider
    .when('/app/main', {
      templateUrl: "template/main.tpl.html",
      controller: "MainCtrl"
    })
    .when('/app/list', {
      templateUrl: "template/list.tpl.html",
      controller: "ListCtrl"
    })
    .when('/app/form', {
      templateUrl: "template/form.tpl.html",
      controller: "FormCtrl"
    })
    .when('/app/network', {
      templateUrl: "template/network.tpl.html",
      controller: "NetworkCtrl"
    })
    .otherwise({redirectTo: '/app/main'});
}]);
```

```
<body ng-app="myApp">
  <div class="body-wrapper">

    <app-header></app-header>

    <div ng-view></div>
  </div>
  <app-footer></app-footer>
</body>
```

ng-init 指令为 AngularJS 应用程序定义了 初始值。但通常情况下使用 **controller** 来设置初始值, 将在下面讨论。

ng-model 指令 绑定 HTML 元素 到应用程序数据。ng-model 通常使用在 input 等输入框的绑定, 通过 ng-model 可以实现双向数据绑定, 下面将详细讨论。

ng-repeat 用于对 HTML 元素的循环, 例如:

```
<div class="page-title">list</div>
<div >
  <div ng-repeat="item in list">{{item.name}}</div>
</div>
```

```
angular.module('listModule', [])
.controller('ListCtrl', ['$scope', function($scope) {
  $scope.list=[{name:"张三"}, {name:"李四"}, {name:"王五"}, {name:"小明"}, {name:"小红"}];
}]);
```

```
list
张三
李四
王五
小明
小红
```

如图: ng-repeat 指令, 循环了 ListCtrl 中 \$scope.list 中的元素, 并将数据绑定到 div 中。

directive 的使用：先通过 js 定义自定义指令，如图

```
/** Created by HF Q on 2016/5/3. ...*/
angular.module('headerModule', [])
.directive('appHeader', function () {
    return {
        restrict: 'EA',
        templateUrl: "template/directiveTpl/header.tpl.html"
    };
})
.controller('pageSwitchCtrl', ['$scope', function($scope) {
    $scope.types=[{name:"main"}, {name:"list"}, {name:"form"}, {name:"network"}];
}])
```

然后创建对应的 html，并将该模块注入到 app 中，注意自定义指令以驼峰命名法，使用时拆分，用-隔开，如<app-header></app-header>，命名时为 appHeader

AngularJS 数据绑定与\$scope, controller

\$scope 即作用域，一般配合 controller 使用，例如

\$scope 的作用范围就是这个 controller 所在元素的范围

\$rootScope 的作用在 ng-app 包含的全局作用域，使用时需注入\$rootScope

如何使用 Scope

当你在 AngularJS 创建控制器时，你可以将 \$scope 对象当作一个参数传递：

AngularJS 实例

控制器中的属性对应了视图上的属性：

```
<div ng-app="myApp" ng-controller="myCtrl">
  <h1>{{carname}}</h1>
</div>

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

app.controller('myCtrl', function($scope) {
    $scope.carname = "Volvo";
});
</script>
```

controller 即控制器，可以控制该控制器所在作用域\$scope 的数据，在 controller 中修改 \$scope.变量的变量值，在 html 中可以用{{变量}}来绑定显示，也可在 html 中用 ng-model 实现双向数据绑定，即 controller 中修改，界面更新，界面中修改，controller 中\$scope 的变量也更新。双向数据绑定常用于有 input 输入的地方。

AngularJS 过滤器

既可过滤显示，也可过滤输入，用于转换数据



AngularJS 过滤器

AngularJS 过滤器可用于转换数据：

过滤器	描述
currency	格式化数字为货币格式。
filter	从数组项中选择一个子集。
lowercase	格式化字符串为小写。
orderBy	根据某个表达式排列数组。
uppercase	格式化字符串为大写。

AngularJS 路由

angularJs 实现路由，需要载入 angular-route.js 文件。在配置路由的模块(app 主模块)中，需要注入 ngRoute。在模块的 config 中，注入\$routeProvider，通过\$routeProvider 的.when() 和.otherwise()方法配置路由。在 html 中使用 ng-view 指令。在文章开始的图中就是样例。

```
$routeProvider.when(url, {  
  template: string,  
  templateUrl: string,  
  controller: string, function 或 array,  
  controllerAs: string,  
  redirectTo: string, function,  
  resolve: object<key, function>  
});
```

如图是路由设置的一些参数。
template 中可直接写 html 元素，
templateUrl 中写自定义的 html 块的 url，controller 是此路由页面的总 controller。controllerAs 为 controller 指定别名。redirectTo 是重定向的地址。resolve 指定当前 controller 所依赖的其他模块

依赖注入

依赖注入，在使用上和 import 有点像，在构建代码时，会有很多的功能，这些功能不能全部写在一个模块或者一个 js 文件当中，常常需要分成很多个模块来实现，遵从单一职责原则，然后需要用到某个模块时再注入这个模块，这个写法和 java 中分成多个类，使用时 import 有点像。

其他常用功能

网络请求，如 get

```
$http.get(url)  
  .success(function(response){  
  })  
  .error(function(error){  
  })
```

表单验证

```
<form name="myForm" novalidate>  
  
  <p>用户名:<br>  
    <input type="text" name="user" ng-model="user" required>  
    <span style="color:red" ng-show="myForm.user.$dirty && myForm.user.$invalid">  
    <span ng-show="myForm.user.$error.required">用户名是必须的。</span>  
  </span>  
  </p>  
  
  <p>邮箱:<br>  
    <input type="email" name="email" ng-model="email" required>  
    <span style="color:red" ng-show="myForm.email.$dirty && myForm.email.$invalid">  
    <span ng-show="myForm.email.$error.required">邮箱是必须的。</span>  
    <span ng-show="myForm.email.$error.email">非法的邮箱。</span>  
  </span>  
  </p>  
  
  <p>  
    <input type="submit"  
      ng-disabled="myForm.user.$dirty && myForm.user.$invalid ||  
      myForm.email.$dirty && myForm.email.$invalid">  
  </p>  
</form>
```



另一个实践地址(多人合作)：<http://139.129.10.20:8080/GHLive/app/index.html#/app/main>