

AngularJS 路由

本章节我们将为大家介绍 AngularJS 路由。

AngularJS 路由允许我们通过不同的 URL 访问不同的内容。

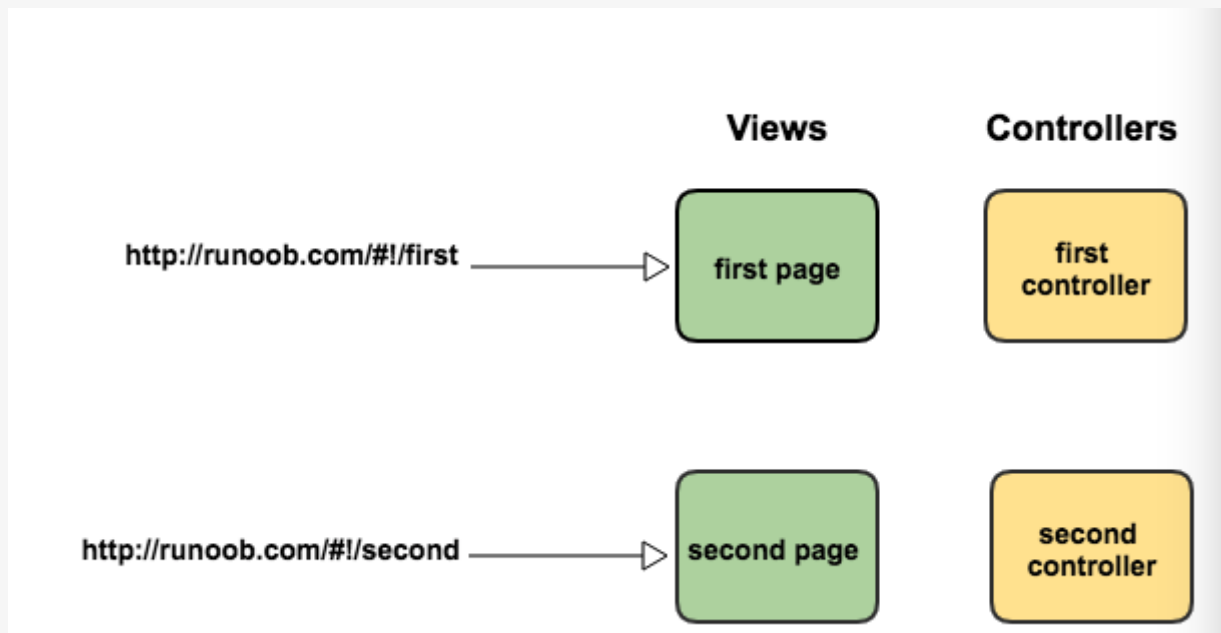
通过 AngularJS 可以实现多视图的单页 Web 应用（single page web application，SPA）。

通常我们的 URL 形式为 `http://runoob.com/first/page`，但在单页 Web 应用中 AngularJS 通过 **#! + 标记** 实现，例如：

```
http://runoob.com/#!/first
http://runoob.com/#!/second
http://runoob.com/#!/third
```

*注意 Angular1.6 之前的版本是通过 **# + 标记** 实现路由。*

当我们点击以上的任意一个链接时，向服务端请求的地址都是一样的 (`http://runoob.com/`)。因为 **#!** 号之后的内容在向服务端请求时会被浏览器忽略掉。所以我们就需要在客户端实现 **#!** 号后面内容的功能实现。AngularJS 路由就通过 **#! + 标记** 帮助我们区分不同的逻辑页面并将不同的页面绑定到对应的控制器上。



在以上图形中，我们可以看到创建了两个 URL：`/ShowOrders` 和 `/AddNewOrder`。每个 URL 都有对应的视图和控制器。

接下来我们来看一个简单的实例：

AngularJS 实例

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>AngularJS 路由实例 - 菜鸟教程</title>
<script src="https://cdn.bootcss.com/angular.js/1.7.0/angular.min.js"></script>
```

```
<script src="https://cdn.bootcss.com/angular.js/1.7.0/angular-route.min.js"></script>
</head>
<body ng-app='routingDemoApp'>
<h2>AngularJS 路由应用</h2>
<ul>
<li><a href="#!/">首页</a></li>
<li><a href="#!/computers">电脑</a></li>
<li><a href="#!/printers">打印机</a></li>
<li><a href="#!/blabla">其他</a></li>
</ul>
<div ng-view></div>
<script>
angular.module('routingDemoApp', ['ngRoute'])
.config(['$routeProvider', function($routeProvider){
$routeProvider
.when('/',{template:'这是首页页面'})
.when('/computers',{template:'这是电脑分类页面'})
.when('/printers',{template:'这是打印机页面'})
.otherwise({redirectTo:'/'});
}]);
</script>
</body>
</html>
```

[尝试一下 »](#)

实例解析：

- 1、载入了实现路由的 js 文件：angular-route.js。
- 2、包含了 ngRoute 模块作为主应用模块的依赖模块。

```
angular.module('routingDemoApp', ['ngRoute'])
```

- 3、使用 ngView 指令。

```
<div ng-view></div>
```

该 div 内的 HTML 内容会根据路由的变化而变化。

- 4、配置 \$routeProvider，AngularJS \$routeProvider 用来定义路由规则。

```
module.config(['$routeProvider', function($routeProvider){
$routeProvider
.when('/',{template:'这是首页页面'})
.when('/computers',{template:'这是电脑分类页面'})
.when('/printers',{template:'这是打印机页面'})
.otherwise({redirectTo:'/'});
}]);
```

AngularJS 模块的 `config` 函数用于配置路由规则。通过使用 `configAPI`，我们请求把 `$routeProvider` 注入到我们的配置函数并且使用 `$routeProvider.whenAPI` 来定义我们的路由规则。

`$routeProvider` 为我们提供了 `when(path,object)` & `otherwise(object)` 函数按顺序定义所有路由，函数包含两个参数：

- 第一个参数是 URL 或者 URL 正则规则。
- 第二个参数是路由配置对象。

路由设置对象

AngularJS 路由也可以通过不同的模板来实现。

`$routeProvider.when` 函数的第一个参数是 URL 或者 URL 正则规则，第二个参数为路由配置对象。

路由配置对象语法规则如下：

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

参数说明：

- **template:**

如果我们只需要在 `ng-view` 中插入简单的 HTML 内容，则使用该参数：

```
.when('/computers',{template:'这是电脑分类页面'})
```

- **templateUrl:**

如果我们只需要在 `ng-view` 中插入 HTML 模板文件，则使用该参数：

```
$routeProvider.when('/computers', {  
  templateUrl: 'views/computers.html',  
});
```

以上代码会从服务端获取 `views/computers.html` 文件内容插入到 `ng-view` 中。

- **controller:**

`function`、`string`或数组类型，在当前模板上执行的`controller`函数，生成新的`scope`。

- **controllerAs:**

string类型，为controller指定别名。

- **redirectTo:**
重定向的地址。
- **resolve:**
指定当前controller所依赖的其他模块。

实例

AngularJS 实例

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>AngularJS 路由实例 - 菜鸟教程</title>
<script src="https://cdn.bootcss.com/angular.js/1.7.0/angular.min.js"></script>
<script src="https://cdn.bootcss.com/angular.js/1.7.0/angular-route.min.js"></script>
<script type="text/javascript">
angular.module('ngRouteExample', ['ngRoute'])
.controller('HomeController', function ($scope, $route) { $scope.$route = $route;})
.controller('AboutController', function ($scope, $route) { $scope.$route = $route;})
.config(function ($routeProvider) {
$routeProvider.
when('/home', {
templateUrl: 'embedded.home.html',
controller: 'HomeController'
}).
when('/about', {
templateUrl: 'embedded.about.html',
controller: 'AboutController'
}).
otherwise({
redirectTo: '/home'
});
});
</script>
</head>
<body ng-app="ngRouteExample" class="ng-scope">
<script type="text/ng-template" id="embedded.home.html">
<h1> Home </h1>
</script>
<script type="text/ng-template" id="embedded.about.html">
<h1> About </h1>
</script>
<div>
<div id="navigation">
<a href="#!/home">Home</a>
<a href="#!/about">About</a>
</div>
<div ng-view="">
</div>
```

```
</div>
</body>
</html>
```

[尝试一下 »](#)[← AngularJS 依赖注入](#)

2 篇笔记

[写笔记](#)

路由设置对象参数规则：

```
$routeProvider.when(url,{
  template:string, //在ng-view中插入简单的html内容
  templateUrl:string, //在ng-view中插入html模版文件
  controller:string,function / array, //在当前模版上执行的controller函数
  controllerAs:string, //为controller指定别名
  redirectTo:string,function, //重定向的地址
  resolve:object<key,function> //指定当前controller所依赖的其他模块
});
```

大兵瑞恩 1年前 (2017-09-30)



根据该教程可知，路由有三种设置。

第一种，设置在 Module 的 Config 阶段，设置 routeProvider，提供简单的 template 模板字符串，当被该路径模式被请求时，将 template 模板字符串渲染在 ng-view 位置。

```
$routeProvider.when('/',{template:'这是首页页面'}).otherwise({redirectTo:'/'});
```

第二种，设置 templateUrl，且页面文件中存在 type为text/ng-template 的模板 js 片段，将会搜索 id 等于 templateUrl 的模板 js，并渲染在 ng-view 的位置。

```
<script type="text/ng-template" id="a.b">...</script>
$routeProvider.when('/',{templateUrl:'a.b'}).otherwise({redirectTo:'/'});
```

第三种，设置 templateUrl，如果页面文件中不存在与 templateUrl 相匹配的 js 模板，则通过相对路径向服务器请求资源，请求完毕再渲染出来。

```
$routeProvider.when('/',{templateUrl:'view/template.html'}).otherwise({redirectTo:'/'});
```

Ar 1个月前 (01-31)