**详解Angular路由 ng-route和ui-router的区别**

**什么是路由？**

路由是AngularJS构建单页面应用的基础。

路由，就是网络数据或者请求进行分发的一个网络组件。

路由就是一个用于请求URL分发和跳转的一个应用组件，Angular中通过$routeProvider路由服务提供者进行核心的配置处理。

**ng路由**

ng 路由是 AngularJS 官方提供的一种简单的路由操作。

ng 路由主要分三个组成部分：路由指令、路由服务、路由服务提供者

**路由指令:ng-view**

ngView指令主要用于将路由指向的页面渲染到当前页面的布局中。

语法：

[?](http://www.jb51.net/article/114350.htm)

|  |  |
| --- | --- |
| 1 | <ng-view [onload=”string”] [autoscroll]=”string”></ng-view> |

1. onload:当视图发生改变时执行属性值中的表达式
2. autoscroll:当视图发生改变时自动触发$anchorScroll事件

事件：

路由视图一旦加载时，就会自动触发$viewContentLoaded 事件

路由提供者：$routeProvider

描述：

内置服务对象 内置服务对象 $routeProvider $routeProvider $routeProvider 主要用于进行路由配置 主要用于进行路由配置  
该服务的使用必须依赖 该服务的使用必须依赖 该服务的使用必须依赖 ngRoute 模块，也就是项目中必须添加 模块，也就是项目中必须添加 animate-route.js

主要方法：

[?](http://www.jb51.net/article/114350.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5 | when( path, route); 用于在访问 path 路径时，跳转到 route 指定的视图    |-- path：路由跳转的径    |-- route：路由对象 <组件 对象 >，一个 JSON对象   otherwise(params);用于在访问不存的路径时，跳转的默认路径或者视图    |-- params：指定路径或者路由对象 |

内置服务<路由服务>：$route &$routeParams

$route 服务被用于进行深层超链接信息的描述， 它会监听 它会监听 $location.url() 地址并进行url 地址和指定的路由视图之间映射关系。

$routeParams服务允许开发人员可以进行路由中参数的处理。

**ui路由**

Angular官方提供的 ng 路由已经具备了非常强大的功能，但是在某些情况下存在一些不太好用的地方，项目中如果出现大量路由深层嵌套话官方的ng路由来处理就不是非常适合了。

第三方提供了一种强大的路由处理功能，对于路由嵌套也可以非常简单的进行处理。

**ui-route：ui路由，第三方路由**

ui路由，主要的功能是项目中的路由操作，和官方的ng路由相比较，它的处理方式更加简洁和易用，尤其是涉及到项目中大量路由嵌套时，使用ui路由能更加快捷方便的完成项目中路由的跳转处理。

**ui-view：指令**

ui -view是 ui 路由中模板页面用于数据展示的指令，可以作为标签的属性出现，同样也可以作为标签的属性出现

[?](http://www.jb51.net/article/114350.htm)

|  |  |
| --- | --- |
| 1  2 | <div ui -viewviewviewview></div>>  <ui -view></uiview> |

同时可以给 ui -view 指令命名，用于在路由跳转时对应名称的 ui -view 上展示对应的数据

**总结：两者区别**

① ng路由实现不了多页面应用

② ui路由：更加方便地处理路由嵌套，在state（）函数中进行即可。

③ngRoute 和 ui-route 相比:

   $when —> $state      路由状态配置的时候

   $routeParams —> $stateParams    带参数的时候的配置

   $routeProvider —> $stateProvider    依赖注入的模块

   <div ng-view></div>  —>  <div ui-view></div>   页面中绑定的指令

④ uiRoute中可以嵌套更深层次的路由，也就是路由中可以有路由（大致有两种情况：1.横向的 2.纵向的）。

　　（1）嵌套路由

　　（2）多视图路由

　　ngRoute中就不能嵌套更深的路由。

**用代码看一下两者的区别：**

ng路由：首先要配置注册 ngRoute

代码如下：

[?](http://www.jb51.net/article/114350.htm)

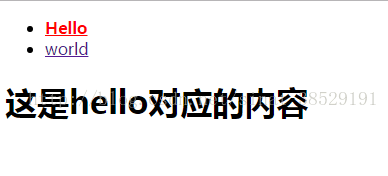
|  |  |
| --- | --- |
| 1  2  3 | var app = angular.module(“myApp", ['ngRoute']);   app.config(["$routeProvider",function($routeProvider){  }]); |

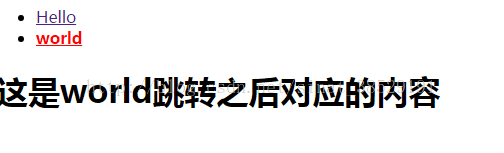
ui路由：用hello world举例子：

[?](http://www.jb51.net/article/114350.htm)

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28 | <!DOCTYPE html>  <html ng-app="myApp">    <head>      <meta charset="UTF-8">      <title>ui-route</title>      <script src="js/lib/angular.min.js" ></script>      <script src="js/lib/angular-ui-router.min.js"></script>    </head>    <style type="text/css">      .active{        color:red;        font-weight:bold;      }    </style>    <body>      <ul>        <li><a ui-sref="hello" ui-sref-active='active'>Hello</a></li>        <li><a ui-sref="world" ui-sref-active='active'>world</a></li>      </ul>      <div>        <ui-view></ui-view>      </div>    </body>    <script>      var app=angular.module("myApp",["ui.router"]);      /\*       $stateProvider: Ui路由中的状态服务提供者      \*/      app.config(["$stateProvider","$urlRouterProvider",function($stateProvider,$urlRouterProvider){        $urlRouterProvider.otherwise("/hello");          $stateProvider.state({          name:"hello",          url:"/hello",          template:"<h1>这是hello对应的内容</h1>"        }).state({          name:"world",          url:"/world",          template:"<h1>这是world跳转之后对应的内容</h1>"        });      }]);      </script>  </html> |

效果如下：





这是简单的一个单页面跳转，有兴趣可以思考一下多页面应用。