عنوان: **یک نکته درباره Angular routeProvider**
نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۲/۱۱/۲۷ ۲۱:۱۰
آدرس: www.dotnettips.info
گروه‌ها: AngularJS

بعد از مطالعه پست‌های ^ و ^ نکته ای به ذهنم رسید که بیان آن از بنده و مطالعه آن توسط شما خالی از لطف نیست. اگر مثال‌های پیاده سازی شده در پست‌های ^ و ^ را با AngularJs نسخه 1.2 اجرا نمایید به طور حتم با خطا روبرو می‌شوید و نتیجه مورد نظر حاصل نمی‌شود. در این پست نیز توسط یکی از دوستان اشاره ای به این مطلب شد.
دلیل خطا این است که از نسخه 1.2 به بعد در Angular سیستم مسیر یابی به این شکل امکان پذیر نیست و بخش مسیریابی به یک فایل دیگر به نام angular-route.js منتقل شده است. در نتیجه اگر به سبک نسخه‌های قبلی Angular از سیستم مسیریابی استفاده نمایید با خطا مواجه خواهید شد و خطای مورد نظر هم مربوط به عدم توانایی در تزریق وابستگی $routeProvider به ماژول مورد نظر است. حال راه حل چیست؟
کافیست در هنگام تعریف ماژول، ngRoute را به عنوان وابستگی ماژول تعیین نمایید. و از طرفی فایل اسکریپتی angular-route.js را بعد از angular.js فراخوانی کنید.
بررسی مثال:
کدهای زیر مربوط به مثال‌های پست قبلی می‌باشد که شرح کامل آن در این پست است:

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

myFirstRoute.config(['$routeProvider',
  function($routeProvider) {
    $routeProvider.
      when('/pageOne', {
        templateUrl: 'templates/page_one.html',
        controller: 'ShowPage1Controller'
      }).
      when('/pageTwo', {
        templateUrl: 'templates/page_two.html',
        controller: 'ShowPage2Controller'
      }).
      otherwise({
        redirectTo: '/pageOne'
      });
}]);


myFirstRoute.controller('ShowPage1Controller', function($scope) {
    $scope.message = 'Content of page-one.html';
});


myFirstRoute.controller('ShowPage2Controller', function($scope) {
    $scope.message = 'Content of page-two.html';
});
```

برای هماهنگ کردن مثال بالا با نسخه 1.2 باید به روش زیر عمل نمود:

```
var myFirstRoute = angular.module('myFirstRoute',['ngRoute']);

myFirstRoute.config(['$routeProvider',
  function($routeProvider) {
    $routeProvider.
      when('/pageOne', {
        templateUrl: 'templates/page_one.html',
        controller: 'ShowPage1Controller'
      }).
      when('/pageTwo', {
        templateUrl: 'templates/page_two.html',
        controller: 'ShowPage2Controller'
      }).
      otherwise({
        redirectTo: '/pageOne'
      });
}]);
```

تنها تغییر، مشخص کردن ngRoute به عنوان وابستگی ماژول myFirstRoute است(خط اول). نکته دیگر لود فایل angular-route.js قبل از فراخوانی فایل بالا و بعد از فراخوانی فایل angular.js است:

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

    <div>
        <div>
        <div>
            <ul>
                <li><a href="#pageOne"> Show page one </a></li>
                <li><a href="#pageTwo"> Show page two </a></li>
            </ul>
        </div>
        <div>
            <div ng-view></div>
        </div>
        </div>
    </div>

    <script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.0.7/angular.min.js"></script>
     <script src="angular-route.js"></script>
    <script src="app.js"></script>

  </body>
```

نظرات خوانندگان

نویسنده: ناصر طاهری
تاریخ: ۱۳۹۲/۱۱/۲۹ ۱۶:۲۶

ممنون از نکته خوبتون.
آیا برای ساماندهی تعداد بالای مسیرها ، راه حلی وجود داره؟

نویسنده: مسعود پاکدل
تاریخ: ۱۳۹۲/۱۱/۲۹ ۲۳:۱۷

بله. من از Angular Dynamic Routing استفاده می‌کنم.

نویسنده: ایاک
تاریخ: ۱۳۹۲/۱۲/۰۱ ۲۰:۸

با سلام.
نظرتون راجع به پروژه angular-ui-router برای مسیریابی انگولار که قابلیت بیشتری را نسبت به سیستم مسیریابی پیش فرض آن ارائه می‌دهد چیست؟

نویسنده: محسن کریمی
تاریخ: ۱۳۹۲/۱۲/۰۲ ۱۳:۲۷

تمام مواردی که در routing مربوط به angularjs هستش در ui-router هم وجود دارد و مواردی مثل nested views و multiple named views بهش اضافه شده و عملا در پروژه‌ها کاربردی‌تر خواهد بود.