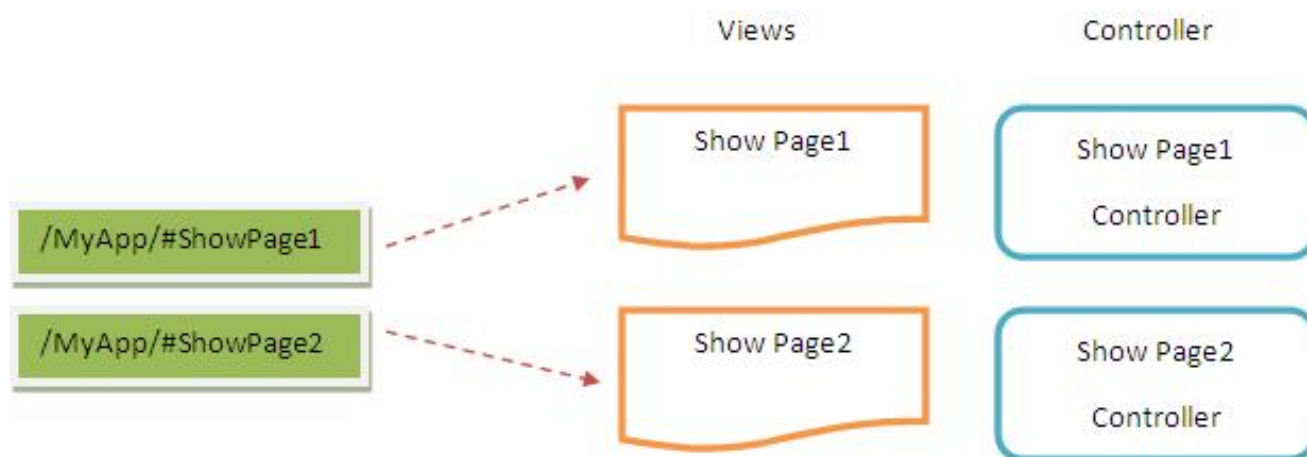


در مطالب قبل کنترلرها و viewها مورد بحث قرار گرفتند. در این پست در نظر داریم یکی از ویژگی‌های دیگر AngularJS به نام مسیریابی (Routing) را مورد بحث قرار دهیم.

یکی از ویژگی‌های برنامه‌های تک صفحه‌ای عدم Reload شدن صفحات است، بر خلاف برنامه‌های وب چند صفحه‌ای که برای نمایش صفحه‌ای دیگر، باید از صفحه‌ای به صفحه‌ای دیگر منتقل شد و عمل Reload هم به طبع نیز اتفاق می‌افتد. در قسمت اول این سری مقالات، مزایای برنامه‌های وب تک صفحه‌ای SPA به صورت کاملتری بیان شده است. در ادامه ما قصد داریم برنامه‌ی وب خود را به صفحات مختلف تقسیم کنیم و سپس با استفاده از امکان مسیریابی موجود در AngularJS آن صفحات را که هر کدام به کنترلری مجزا مقید شده‌اند، در صفحه‌ی اصلی خود بارگزاری کنیم. همچنین استفاده از مسیریابی موجود، میتواند به ما در مدیریت بهتر صفحات کمک فراوانی بکند. به تصویر زیر دقت کنید:



در تصویر بالا دو مسیر با آدرس‌های /ShowPage1 و /ShowPage2 تعریف شده است که هر کدام به یک view مشخص و یک Controller برای مدیریت آن اشاره میکند.

زمانی که ما از تزریق وابستگی‌ها در AngularJS استفاده میکنیم و یک شیء را به کنترلر تزریق میکنیم، Angular توسط \$Injector سعی در پیدا کردن وابستگی مربوطه و سپس تزریق آن به کنترلر را انجام میدهد. برای استفاده از امکان مسیریابی Route، ما نیز باید از پروایدر مخصوص آن برای تزریق استفاده کنیم. در Angular مسیرهای برنامه توسط پروایدری به نام \$routeProvider شناسایی میشود که خدمات مسیریابی را به ما ارائه میدهد. این سرویس به ما کمک میکند تا بتوانیم اتصال بین کنترلرها، ویوها و آدرس URL جاری مرورگرها را به آسانی برقرار کنیم.

بهتر است کار را شروع کنیم. یک فایل JS ایجاد و سپس محتویات زیر را در آن قرار دهید:

```
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';
});

```

در کدهای بالا ابتدا یک ماژول تعریف کرده ایم و سپس توسط `config()` تنظیمات مربوط به مسیریابی را انجام داده ایم. با استفاده از متدهای `when` و `otherwise` میتوانیم مسیرها را تعریف کنیم. برای هر مسیر دو پارامتر وجود دارد که اولین پارامتر نام مسیر و دومین پارامتر شامل 2 قسمت میشود که `templateUrl` آن آدرسی که باز خواهد شد و `controller` نیز نام کنترلری که ویو را مدیریت میکند.

توسط `otherwise` میتوانیم مسیر پیشفرض را نیز تعریف کنیم تا در صورتی که مسیری با آدرس‌های بالای آن مطابقت نداشت به این آدرس منتقل شود.

در قطعه کد بالا همچنین دو مسیر با نام‌های `/pageOne` و `/pageTwo` تعریف کرده ایم که هر کدام به ترتیب به `View`های : `templates/page_one.html` و `templates/page_two.html` مرتبط شده اند. همچنین دو کنترلر برای مدیریت ویوها نیز تعریف شده است.

زمانی که ما آدرس `http://appname/#pageOne` را در نوار آدرس مرورگر وارد میکنیم، Angular به صورت اتوماتیک آدرس URL را با تنظیماتی که ما در اینجا تعریف کرده ایم مطابقت میدهد و در صورت وجود چنین آدرسی، `view` مربوطه را بارگذاری میکند و در این مثال نیز مطابق با تنظیمات بالا، صفحه‌ی `templates/page_one.html` برای ما بارگذاری و سپس کنترلر `ShowPage1Controller` را فراخوانی میکند، جایی که ما منطق کار را در آن قرار میدهیم.

محتویات فایل `main.html` :

```

<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="app.js"></script>

</body>

```

در قطعه کد بالا دو لینک تعریف شده است که ویژگی `href` از علامت هش و نام صفحه تشکیل شده است. یکی از چیزهایی که

شایان ذکر است ، دایرکتیو ng-view است. مکانی برای بارگذاری صفحات در آن.

ما میتوانیم این تگ را به سه شکل زیر نیز استفاده کنیم :

```
<div ng-view></div>
..
<ng-view></ng-view>
..
<div class="ng-view"></div>
```

محتویات صفحه templates/page_one.html :

```
<h2>Page One</h2>
{{ message }}
```

محتویات صفحه templates/page_two.html :

```
<h2>Page Two</h2>
{{ message }}
```

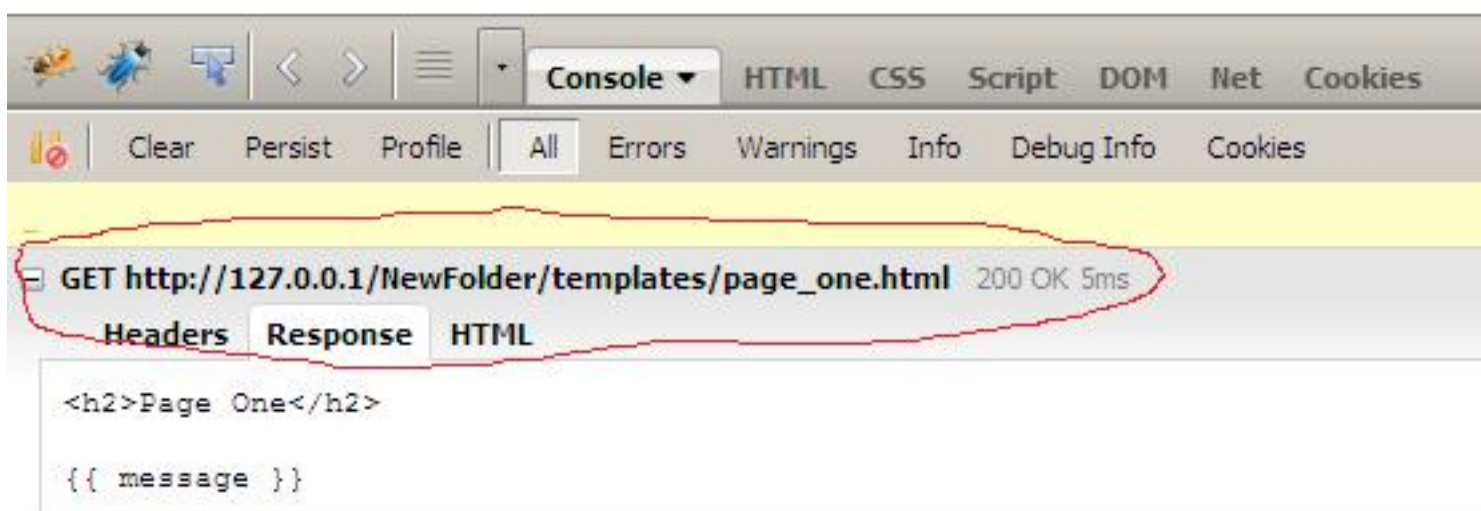
حال اگر پروژه را اجرا کنید و به کنسول مرور گر خود نگاه کنید متوجه میشوید که مسیریاب از مسیر پیشفرض استفاده کرده است و صفحه‌ی page_one.html را به صورت ایجکسی فراخوانی کرده است :

127.0.0.1/NewFolder/#/pageOne

- [Show page one](#)
- [Show page two](#)

Page One

Content of page-one.html



و اگر روی لینک Show Page two کلیک کنید ، صفحه‌ی page_two.html نیز به صورت ایجکسی فراخوانی میشود.

دوباره بر روی لینک Show page one کلیک کنید. بله. هیچ درخواستی به سمت سرور ارسال نشد و صفحه‌ی page_one.html به خوبی نمایش داده شد. یکی از مزیت‌های سیستم مسیریابی قابلیت کش کردن صفحات است تا در صورت فراخوانی مجدد، درخواستی به سمت سرور ارسال نشود و خیلی سریع به شما نمایش داده شود.

مثال این مطلب : [RouteExample.zip](#)

ادامه دارد ...

نظرات خوانندگان

نویسنده: حسین
تاریخ: ۱۳۹۲/۰۹/۲۵ ۱۲:۵۹

سلام این نمونه کد در فایرفاکس به درستی کار میکنه ولی در گوگل کروم یا ie با خطاهای زیر مواجه میشم

```
1)OPTIONS file:///E:/Users/admin/Downloads/Compressed/RouteExample/RouteExample/templates/page_two.html
No 'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'null' is
therefore not allowed access. angular.min.js:99
```

```
2)XMLHttpRequest cannot load
file:///E:/Users/admin/Downloads/Compressed/RouteExample/RouteExample/templates/page_two.html. No
'Access-Control-Allow-Origin' header is present on the requested resource. Origin 'null' is therefore
not allowed access.
```

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۰۹/۲۵ ۱۳:۳۴

برای اجرا نیاز به سرور داره. ولی کروم برای تست با دستور خط فرمان Chrome.exe --allow-file-access-from-files چنین اجازه‌ای رو به شما میده.

نویسنده: ناصر طاهری
تاریخ: ۱۳۹۲/۰۹/۲۵ ۱۳:۳۴

پروژه را تحت یک سرور مثل Apache یا IIS اجرا کنید. مشکل رفع میشود.

نویسنده: وحید م
تاریخ: ۱۳۹۲/۱۰/۰۵ ۲۳:۵۰

در mvc چطور میشود از حالت روتینگ استفاده کرد بگونه ای که مثلا بجای فایل html بتوان فایل cshtml جهت لود کردن partial view استفاده کرد. و اینکه چگونه مسیر دهی مثل views/home/_partial.cshtml داشته باشیم ممنون از شما

نویسنده: محسن خان
تاریخ: ۱۳۹۲/۱۰/۰۶ ۰:۶

نمیشه مستقیما. چون مسیر کاری MVC از اکشن متد و کنترلرها شروع میشه و نه از View ها. View های قرار گرفته در پوشه Views دارای web.config عدم دسترسی از خارج از سایت هستند. کدهای angularjs هم سمت کاربر هست و نه سمت سرور.

نویسنده: ناصر طاهری
تاریخ: ۱۳۹۲/۱۰/۰۶ ۰:۷

در این [کامنت](#) هم توضیحاتی داده شده.

نویسنده: سعید رضایی
تاریخ: ۱۳۹۳/۰۲/۱۴ ۱۳:۴۶

به جای page_one.html چجوری میشه از cshtml استفاده کرد.
کلا کار درسته که از cshtml فایل‌ها به templateUrl استفاده کرد یا نه؟
با تشکر

نویسنده: محسن درپرستی
تاریخ: ۱۸:۱ ۱۳۹۳/۰۲/۱۴

اگر بخواید از فایل‌های cshtml استفاده کنید باید از طریق یک اکشن باید اینکار رو بکنید. درست یا غلط بودنش به سناریویی که دارید بستگی دارد.

نویسنده: ناصر طاهری
تاریخ: ۱۸:۵۶ ۱۳۹۳/۰۲/۱۴

یک مثال هم در این کامنت قرار داده شده است. ([+](#))