

در پست قبلی با مفاهیم کنترلر و مدل در AngularJS آشنا شدید. قصد دارم روشی را بررسی کنم که یک منبع داده را بین کنترلرهای تعریف شده در یک ماژول را به اشتراک بگذاریم.

ابتدا یک فایل جاوااسکریپت به نام module1 ایجاد می‌کنیم. در این فایل ابتدا ماژول خود را به Angular معرفی کرده و سپس با استفاده از دستور factory سرویس مورد نظر برای به اشتراک گذاری داده را می‌سازیم:

```
var app = angular.module('myApp', []);
app.factory('BookData', function () {
    var books = [
        { code: 1, name: 'book1', },
        { code: 2, name: 'book2', },
        { code: 3, name: 'book3', },
        { code: 4, name: 'book4', },
        { code: 5, name: 'book5', },
    ];
    return books;
});
```

همان طور که در پست قبلی شرح داده شده برای تعریف ماژول از دستور angular.module استفاده می‌کنیم. در خط بعدی یک سرویس به نام BookData را با استفاده از دستور factory در ماژول مربوطه ساخته می‌شود. تابع مورد نظر بک آرایه از کتاب‌ها را که هر کدام از آن‌ها شامل کد و نام است برگشت می‌دهد. قصد داریم کنترلرهای تعریف شده در ماژول myApp بتوانند به این لیست این کتاب‌ها دسترسی داشته باشند. در این مرحله ابتدا یک کنترلر به نام controller1 به صورت زیر می‌سازیم:

```
app.controller('controller1', function ($scope, BookData) {
    $scope.books = BookData;
});
```

تنها نکته قابل ذکر، تزریق مقادیر \$scope و BookData به تابع سازنده کنترلر مربوطه است. از \$scope برای مقید سازی مقادیر مدل به عناصر dom در view استفاده می‌شود و BookData در این جا دقیقا به مقدار برگشت داده شده از سرویس BookData اشاره می‌کند (نام سرویس مورد نظر دقیقا باید با مقداری که به عنوان آرگومان اول در تابع factory پاس می‌دهید یکی باشد). در نتیجه این مقدار را به متغیر books در \$scope نسبت می‌دهیم. برای کنترلر دوم نیز همین مراحل را تکرار می‌کنیم:

```
app.controller('controller2', function ($scope, BookData) {
    $scope.books = BookData;
});
```

در View مورد نظر نیز یک ارجاع به فایل ساخته شده بالا خواهیم داشت و سپس کدهای مربوط به نمایش را به صورت زیر می‌نویسیم (البته ارجاع به فایل اصلی angular.js فراموش نشود):

```
<script type="text/javascript" src="~/scripts/app/controller1.js"></script>

<div ng-app="myApp">
  <div ng-controller="controller1">
    <p>Data from controller1</p>
    <table>
      <tr ng-repeat="book in books">
        <td>
          {{book.code}}
        </td>
        <td>
          {{book.name}}
        </td>
      </tr>
    </table>
  </div>
</div>
```

```
        </tr>
      </table>
    </div>

    <div ng-controller="controller2">
      <p>Data from controller2</p>
      <table>
        <tr ng-repeat="book in books">
          <td>
            {{book.code}}
          </td>
          <td>
            {{book.name}}
          </td>
        </tr>
      </table>
    </div>
  </div>
```

ابتدا در تگ div اول با استفاده از ng-app محدودۀ ماژول مورد نظر در صفحۀ را تعیین کرده سپس با استفاده از تگ‌های div جداگانه هر کدام از نواحی تحت کنترل مربوط به کنترلرهای تعریف شده را مشخص می‌کنیم. با استفاده از ng-repeat به راحتی در بین آرایه کتاب‌ها پیمایش کرده و لیست مورد نظر در صفحۀ نمایش داده می‌شود. (توضیحات مربوط به ng-repeat و {{}} در [پست قبلی](#) شرح داده شده است). خروجی به صورت زیر خواهد بود. واضح است که اطلاعات نمایش داده شده توسط هر دو کنترلر به دلیل استفاده از منبع داده ای یکسان، به یک شکل خواهد بود.

Data from controller1

1	book1
2	book2
3	book3
4	book4
5	book5

Data from controller2

1	book1
2	book2
3	book3
4	book4
5	book5

نظرات خوانندگان

نویسنده: محسن خان
تاریخ: ۹:۵۷ ۱۳۹۲/۰۹/۱۷

ممنون از شما. با توجه به اینکه در factory ایجاد شده یک نمونه جدید از books ایجاد می‌شود، آیا با تزریق آن در سازنده‌های کنترلرهای مختلف، اگر تغییری نیز در یک کنترلر بر روی این books انجام شود، در بقیه هم منعکس می‌شود؟

نویسنده: ناصر طاهری
تاریخ: ۱۰:۴۳ ۱۳۹۲/۰۹/۱۷

بله. اگر متدی را به کنترلر 2 اضافه کنید و سپس یک تگ لینک در view برای فراخوانی این متد، به صورت زیر :

```
sampleApp.controller('controller2', function ($scope, BookData) {  
    $scope.books = BookData;  
    $scope.change = function(){  
        BookData[0].name = 'change this book';  
    }  
});
```

و view :

```
<div ng-controller="controller2">  
    <p>Data from controller2</p>  
    <table>  
        <tr ng-repeat="book in books">  
            <td>  
                {{book.code}}  
            </td>  
            <td>  
                {{book.name}}  
            </td>  
        </tr>  
    </table>  
    <a href="javascript:void(0)" ng-click="change();" >Click here to change book1</a>  
</div>
```

میبینید که در کنترلر شماره 1 هم تغییر میکند.