

در این مقاله قصد داریم تا به بررسی و پیاده سازی کنترلرهای تو در تو در AngularJS بپردازیم. به این صورت که میتوانیم در یک صفحه یک کنترلر اصلی را در نظر بگیریم و کنترلرهای دیگر را در این کنترلر قرار دهیم. نحوه ی ارث بری متغیرها، انقیاد داده ها و مقادیر تعریف شده در \$scope، از جمله مواردی است که به آن خواهیم پرداخت. تمامی مواردی که ذکر خواهد شد در قالب یک پروژه قرار داده شده است. [AngularJsNestedController.zip](#)

حال به بررسی جزئی کنترلرهای تودرتو یا همان Nested Controllers می پردازیم. در این پروژه قصد داریم تا با تعریف سه متغیر firstName و lastName و middleName نحوه ارث بری پارامترها را در این کنترلرهای تو در تو بررسی نماییم. ساختار یک صفحه با کنترلرهای تو در تو به صورت زیر می باشد. همانطور که مشاهده میکنید کنترلر دوم و سوم به صورت nested درون کنترلر اول تعریف شده اند.

```
<body ng-app = "app">
  <div ng-controller="firstControllerScope">
    <div ng-controller="secondControllerScope">
      <div ng-controller="thirdControllerScope">
      </div>
    </div>
  </div>
</body>
```

در کدهای پیاده سازی شده این کنترلر به صورت زیر می باشد. همانطور که مشاهده میکنید در کنترلر اول تنها پارامتر firstName و در کنترلر دوم و سوم نیز به ترتیب lastName و middleName اضافه شده اند. همچنین دو تابع utility به نام getFullName در کنترلر دوم و سوم تعریف شده است.

```
var firstControllerScope = function ($scope) {
  // Initialize the model variables
  $scope.firstName = "John";
};

var secondControllerScope = function ($scope) {
  // Initialize the model variables
  $scope.lastName = "Doe";

  // Define utility functions
  $scope.getFullName = function () {
    return $scope.firstName + " " + $scope.lastName;
  };
};

var thirdControllerScope = function ($scope) {
  // Initialize the model variables
  $scope.middleName = "Al";
  $scope.lastName = "Smith";

  // Define utility functions
  $scope.getFullName = function () {
    return $scope.firstName + " " + $scope.middleName + " " + $scope.lastName;
  };
};
```

فایل html حاوی این کنترلرها در زیر نمایش داده شده است. همانطور که مشاهده میکنید تمامی کنترلر دوم یعنی secondControllerScope و کنترلر سوم درون firstControllerScope تعریف شده اند. همچنین دو تابع utility با نامهای getFullName نیز در زیر مقدار fullName را باز میگردانند.

```
<div ng-controller="firstControllerScope">
  <h3>First controller</h3>
  <strong>First name:</strong> {{firstName}}<br />
  <br />
  <label>first name: <input type="text" ng-model="firstName" /></label><br />
  <br />
```

```
<div ng-controller="secondControllerScope">
  <h3>Second controller (inside First)</h3>
  <strong>First name (from First):</strong> {{firstName}}<br />
  <strong>Last name (new variable):</strong> {{lastName}}<br />
  <strong>Full name:</strong> {{getFullName()}}<br />
  <br />
  <label>first name: <input type="text" ng-model="firstName" /></label><br />
  <label>last name: <input type="text" ng-model="lastName" /></label><br />
  <br />

  <div ng-controller="thirdControllerScope">
    <h3>Third controller (inside Second and First)</h3>
    <strong>First name (from First):</strong> {{firstName}}<br />
    <strong>Middle name (new variable):</strong> {{middleName}}<br />
    <strong>Last name (from Second):</strong> {{$parent.lastName}}<br />
    <strong>Last name (redefined in Third):</strong> {{lastName}}<br />
    <strong>Full name (redefined in Third):</strong> {{getFullName()}}<br />
    <br />
    <label>first name: <input type="text" ng-model="firstName" /></label><br />
    <label>middle name: <input type="text" ng-model="middleName" /></label><br />
    <label>last name: <input type="text" ng-model="lastName" /></label>
  </div>
</div>
</div>
```

با توجه به مقادیر تعریف شده درون هر یک از کنترلرها و با اجرای پروژه نمونه مشاهده میکنیم که اگر یک پارامتر مدل (مانند middleName) را درون کنترلر فرزند به وسیله input fieldها تغییر دهیم، این تغییر به والد منتقل نمیشود. یعنی اگر به عنوان مثال ما lastName را در کنترلر سوم تغییر دهیم دیگر lastName که در کنترلر سوم است تغییر نخواهد کرد. حال اگر همان lastName در کنترلر دوم که در یک فیلد input قرار داده شده تغییر کند، مشاهده میکنیم که در کنترلر سوم نیز مقدار lastName تغییر خواهد کرد.

در بخش بعدی نوع دیگری از ارث بری پارامترها و تابعها در کنترلرهای AngularJs را شرح خواهیم داد.