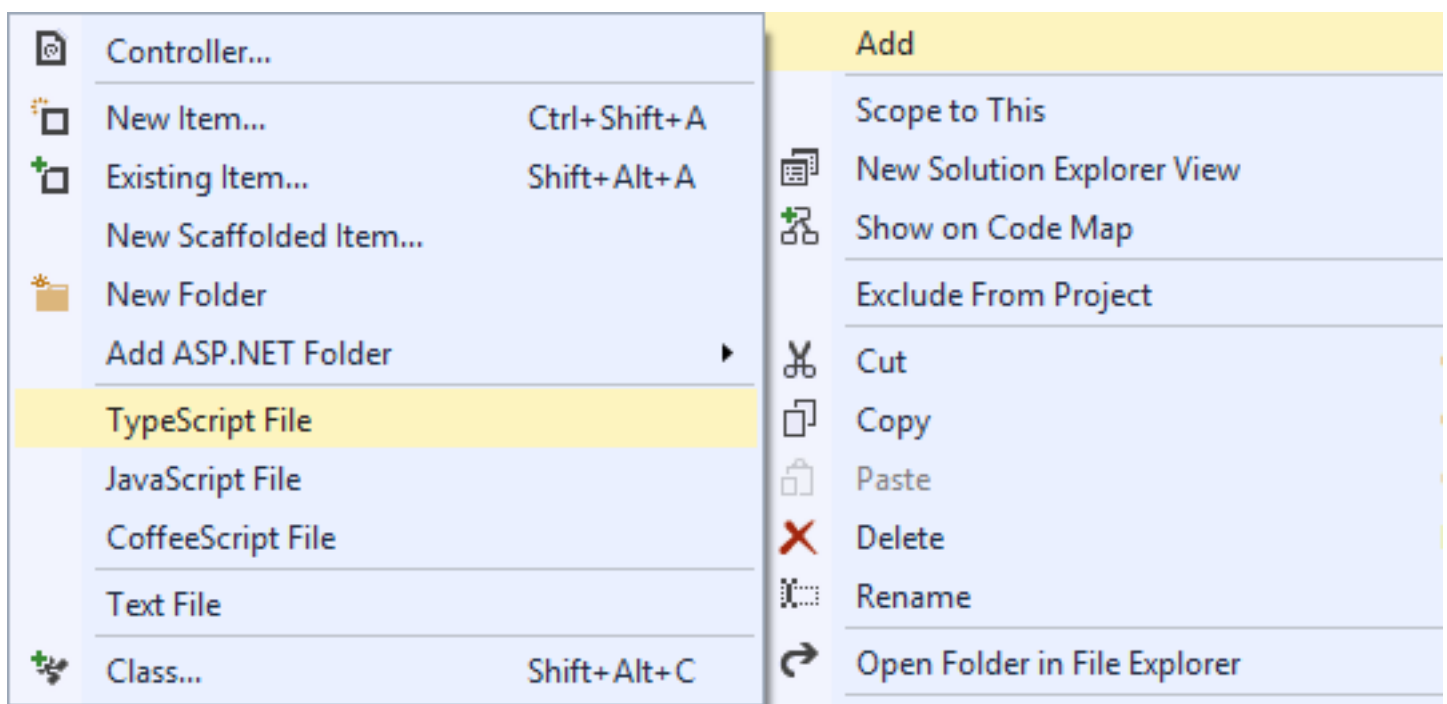


پیشتر با ویژگی ها و نحوه کد نویسی این زبان آشنا شدید. از طرفی دیگر، نحوه تعریف کنترلرها در Angular نیز آموزش داده شد. در این پست قصد داریم طی یک مثال ساده با استفاده از زبان Typescript یک کنترلر Angular را ایجاد و سپس از آن در یک پروژه Asp.Net MVC استفاده نمایم. از آن جا که به صورت پیش فرض در VS.Net امکانات TypeScript نصب نشده است، برای شروع ابتدا TypeScript را از اینجا دانلود نمایید. بعد از نصب یک پروژه Asp.Net MVC ایجاد نمایید و سپس با استفاده از nuget فایل‌های مربوط به AngularJs را نصب نمایید. در این پست به تفصیل این مورد بررسی شده است (عملیات BundleConfig فایل‌های مورد نیاز به عهده خودتان). در پوشه scripts یک فولدر به نام app ساخته، سپس یک فایل TypeScript به نام ProductController.ts ایجاد کنید. (بعد از نصب TypeScript گزینه TypeScript File مشاهده خواهد شد)



در فایل ProductController.ts کدهای زیر را کپی نمایید:

```
module Product {
    export interface Scope {
        message: string;
    }

    export class Controller {
        constructor($scope: Scope) {
            $scope.message = "Hello from Masoud";
        }
    }
}
```

توضیح کدها بالا :

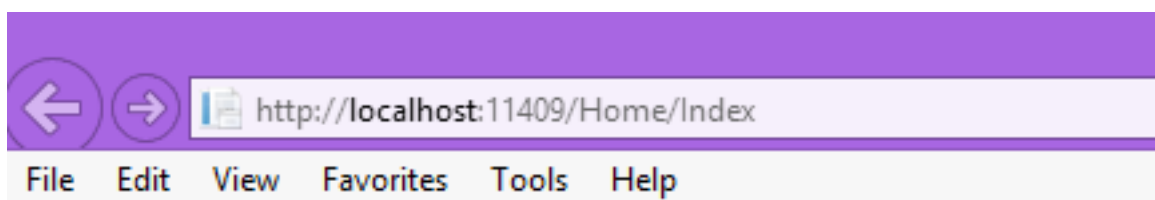
ابتدا یک ماژول به نام Product ایجاد می‌کنیم. سپس یک اینترفیس برای ایجاد Scope که جهت مقید سازی عناصر DOM به آبجکت‌های کنترلر مورد استفاده قرار می‌گیرد، ایجاد می‌کنیم. در داخل این اینترفیس متغیری به نام message از نوع string داریم. قصد داریم این متغیر را به یک عنصر مقید کنیم. حال یک کلاس به نام کنترلر ایجاد می‌کنیم که در تابع سازنده آن تزریق

وابستگی برای `$scope` از نوع اینترفیس `Scope` تعیین شده است. در نتیجه در بدنه سازنده می‌توانیم به متغیر `message` مقدار مورد نظر را نسبت دهیم.

کلمه کلیدی `export` برای تعریف عمومی کلاس استفاده شده است. یک `View` ایجاد و کدهای زیر را در آن کپی کنید:

```
<script type="text/javascript" src="~/scripts/app/ProductController.js"></script>
<div ng-app>
  <div ng-controller="Product.Controller">
    <p>{{message}}</p>
  </div>
</div>
```

اولین نکته در تگ `script` است که فراخوانی فایل `Typescript` باید با پسوند `.js` انجام گیرد. به دلیل اینکه فایل‌های `Typescript` بعد از کامپایل تبدیل به فایل‌های `JavaScript` خواهند شد؛ در نتیجه پسوند آن نیز `.js` است. دومین نکته در فراخوانی کنترلر مورد نظر است که از ترکیب نام ماژول و نام کلاس است. بعد از اجرای پروژه خروجی به صورت زیر خواهد بود:



نظرات خوانندگان

نویسنده: ایلیا اکبری فرد
تاریخ: ۱۵:۲۶ ۱۳۹۲/۱۰/۲۵

با سلام.
در کنترلر ، چگونه watch\$ مربوط به شی scope\$ را بوسیله TypeScript میتوان فراخوانی کرد؟

نویسنده: مسعود پاکدل
تاریخ: ۱۳:۷ ۱۳۹۲/۱۰/۲۷

ابتدا کنترلر خود را به صورت زیر تعریف کنید:

```
class MyController {  
    thescope: any;  
    static $inject = ['$scope'];
```

```
    constructor($scope) {  
        this.thescope = $scope;
```

```
        this.thescope.$watch('watchtext', function(newValue, oldValue) {  
            this.thescope.counter = scope.counter + 1;  
            this.thescope.lastvalue = oldValue;  
            this.thescope.currentvalue = newValue;});
```

```
    }  
}
```

حال برای استفاده از کنترلر بالا به صورت زیر عمل نمایید:

```
module myApp.ctrl1 {  
    myApp.controller("MyController", function($scope) {  
        return new MyController($scope);  
    })  
}
```