Integrar Angular con ASP.NET MVC

Introducción:

Este artículo mostrará la configuración básica de angular integrada con ASP.Net MVC.

Prerrequisitos:

Instala Visual Studio 2015 (incluso puedes usar 2013). Haga clic aquí para descargar

Instale el paquete Node js para Visual Studio. Haga clic aquí para descargar

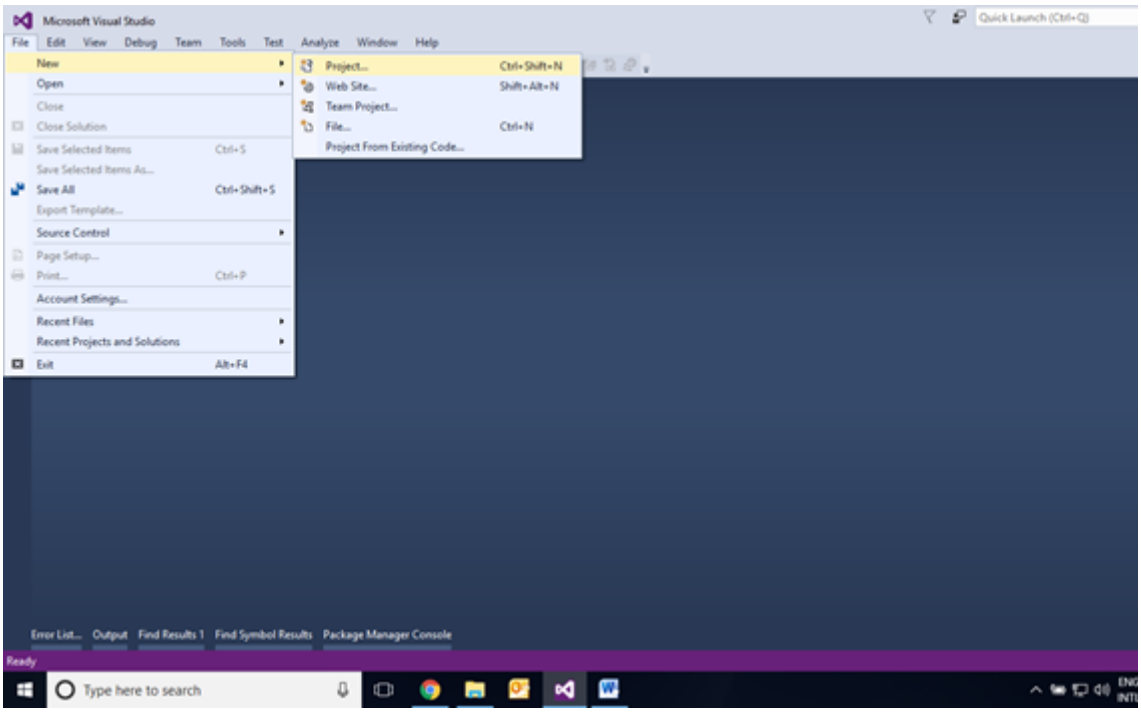
Instalar el paquete de Script para Visual Studio. Haga clic aquí para descargar

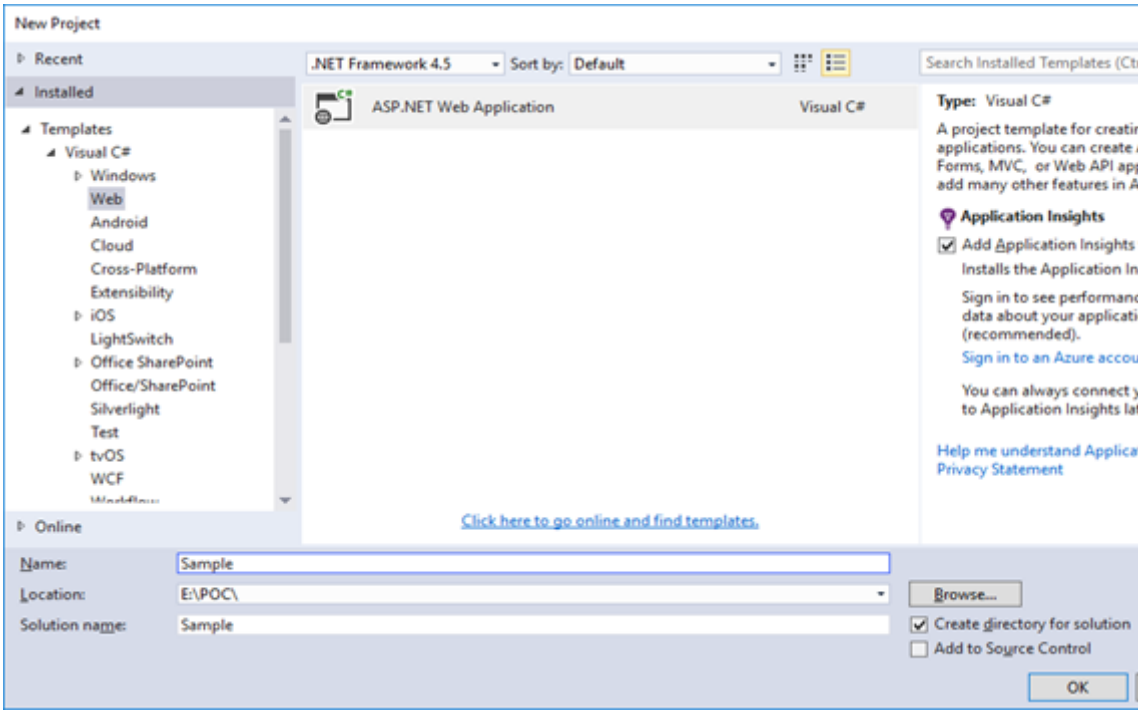
Reader debería tener conocimientos básicos sobre arquitectura MVC y Angular 2 para una mejor comprensión.

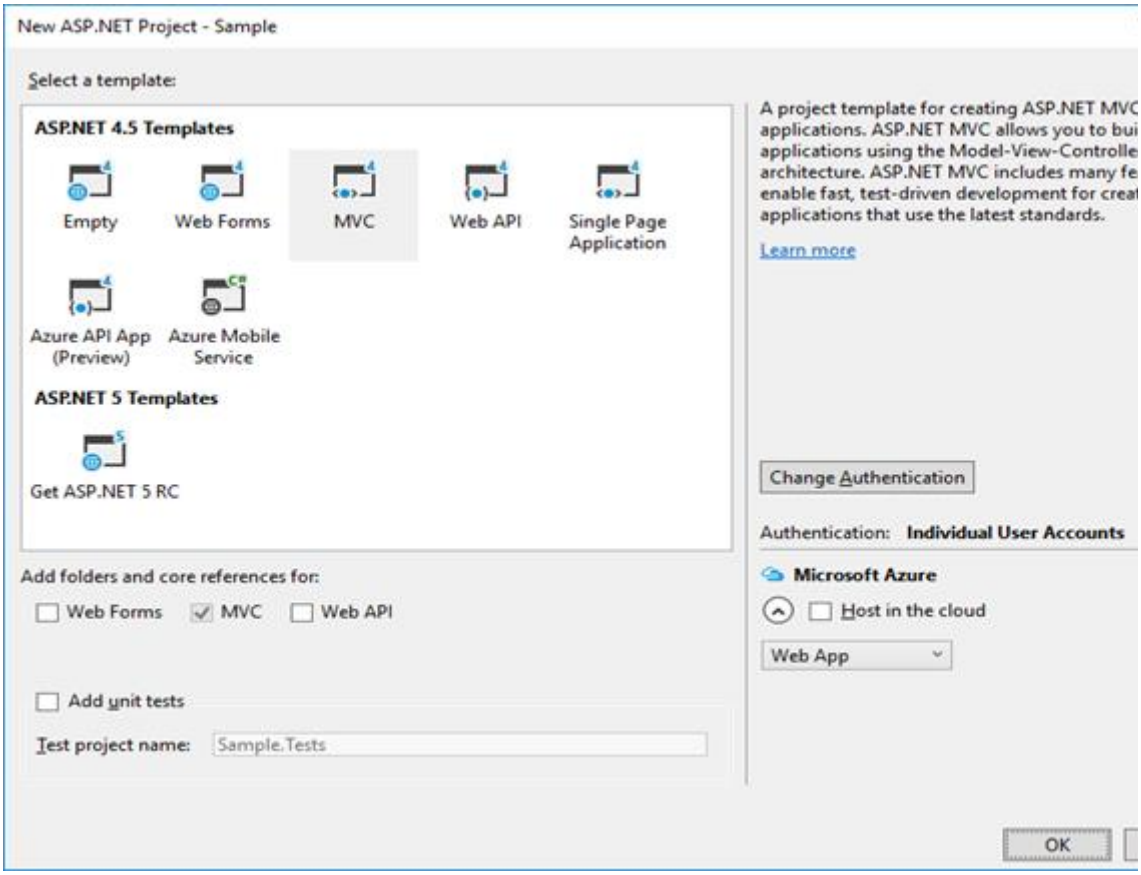
Cree una aplicación MVC de muestra con angular:

Paso 1: Agregar nuevo proyecto MVC

- Crea un nuevo proyecto con los pasos a continuación.



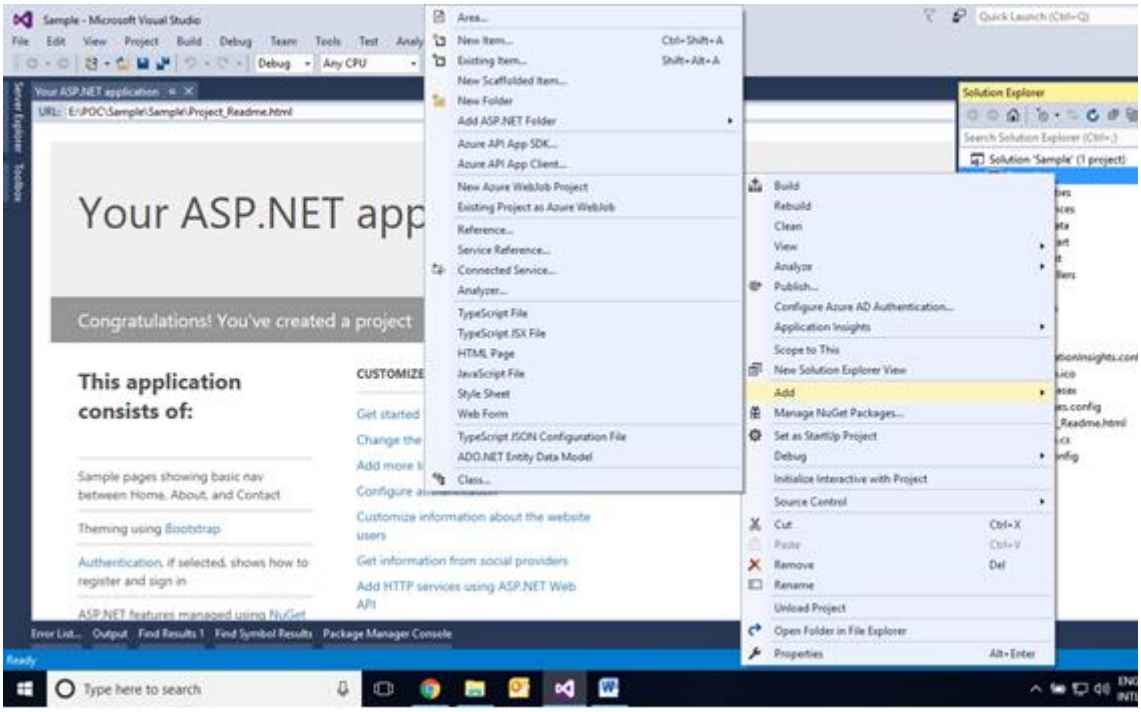

New Project
Recent
Installed
Templates
Visual C#
Windows
Web
Android
Cloud
Cross-Platform
Extensibility
iOS
LightSwitch
Office SharePoint
Office/SharePoint
Silverlight
Test
tvOS
WCF
Online
.NET Framework 4.5
Sort by: Default
Search Installed Templates (Ct
ASP.NET Web Application
Visual C#
Type: Visual C#
A project template for creati
applications. You can create
Forms, MVC, or Web API ap
add many other features in A
Application Insights
Add Application Insights
Installs the Application In
Sign in to see performan
data about your applicati
(recommended).
Sign in to an Azure accou
You can always connect
to Application Insights la
Help me understand Applica
Privacy Statement
Click here to go online and find templates.
Name:
Sample
Location:
E:\POC\
Solution name:
Sample
Browse...
Create directory for solution
Add to Source Control
OK


New ASP.NET Project - Sample
Select a template:
ASP.NET 4.5 Templates
Empty
Web Forms
MVC
Web API
Single Page Application
Azure API App (Preview)
Azure Mobile Service
ASP.NET 5 Templates
Get ASP.NET 5 RC
A project template for creating ASP.NET MVC
applications. ASP.NET MVC allows you to bui
applications using the Model-View-Controlle
architecture. ASP.NET MVC includes many fe
enable fast, test-driven development for creat
applications that use the latest standards.
Learn more
Change Authentication
Authentication: Individual User Accounts
Microsoft Azure
Host in the cloud
Web App
Add folders and core references for:
Web Forms
MVC
Web API
Add unit tests
Test project name:
Sample.Tests
OK

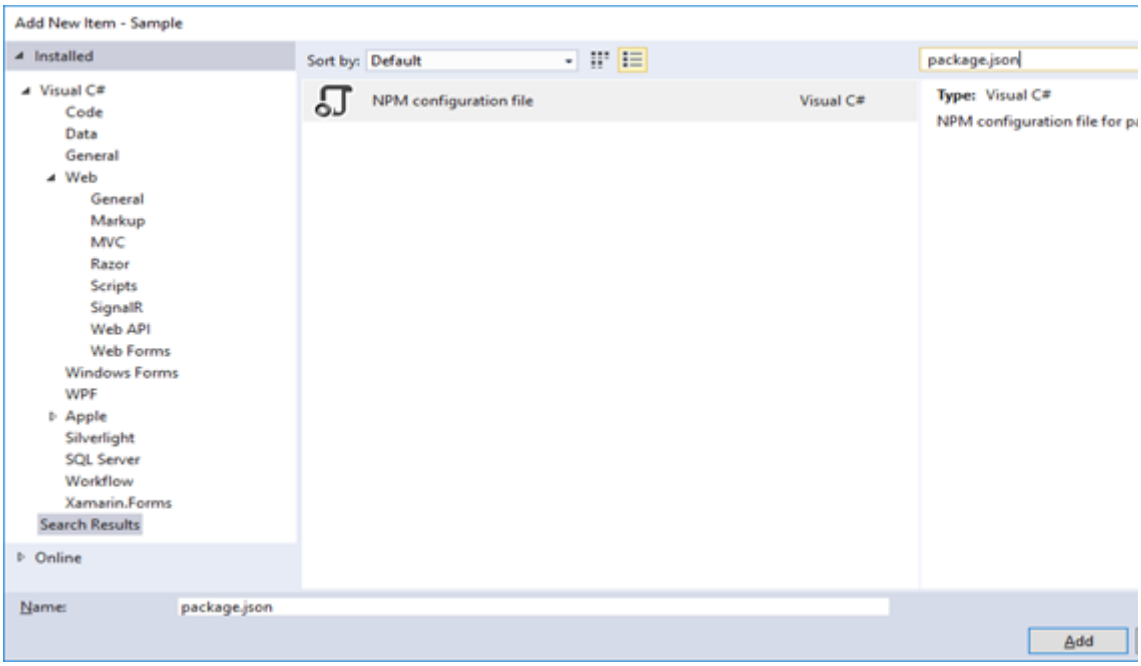
Paso 2: agregue un archivo de paquete para angular

- Agregue un archivo package.json para importar las referencias requeridas con los pasos a continuación.

- Haga clic derecho en el proyecto, Agregar → Nuevo elemento



- Buscar con package.json y Agregar.



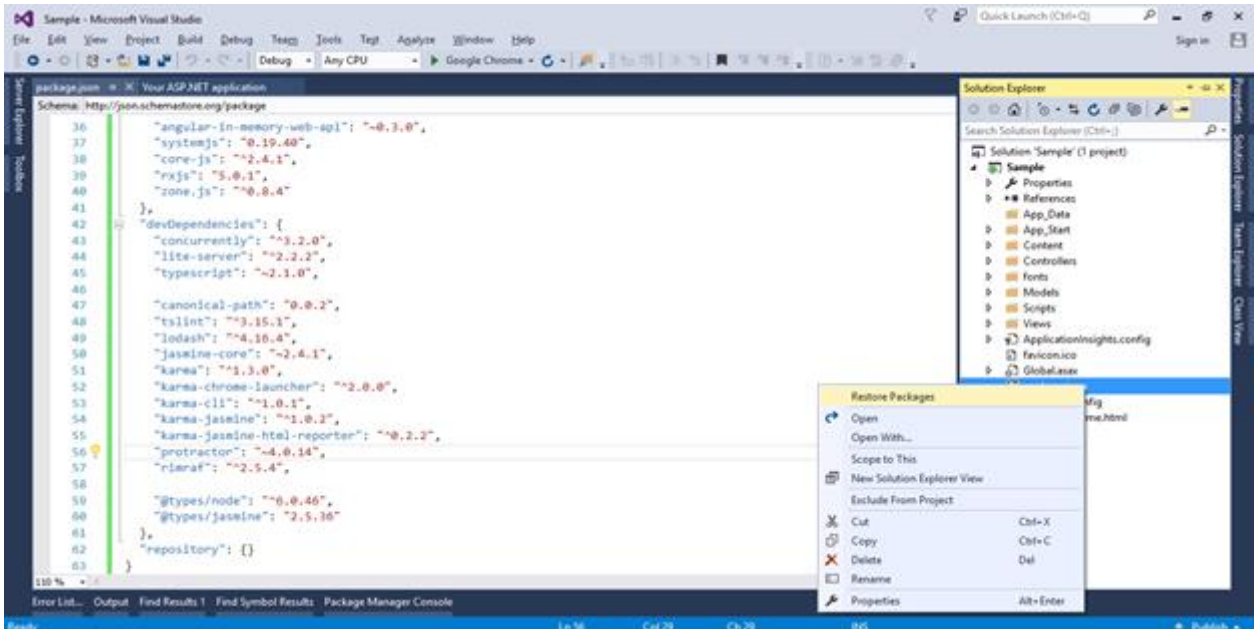
•Pegue el fragmento de código a continuación en el archivo package.json creado.

```
1. {
2.
3.   "name": "angular-quickstart",
4.
5.   "version": "1.0.0",
6.
7.   "description": "QuickStart package.json from the
   documentation, supplemented with testing support",
8.
9.   "scripts": {
10.
11.         "build": "tsc -p src/",
12.
13.         "build:watch": "tsc -p src/ -w",
14.
15.         "build:e2e": "tsc -p e2e/",
16.
17.         "serve": "lite-server -c=bs-config.json",
18.
19.         "serve:e2e": "lite-server -c=bs-config.e2e.json",
20.
21.         "prestart": "npm run build",
22.
23.         "start": "concurrently \"npm run build:watch\" \"npm
   run serve\"",
24.
25.         "pree2e": "npm run build:e2e",
26.
27.         "e2e": "concurrently \"npm run serve:e2e\" \"npm run
   protractor\" --kill-others --success first",
28.
29.         "preprotractor": "webdriver-manager update",
30.
```

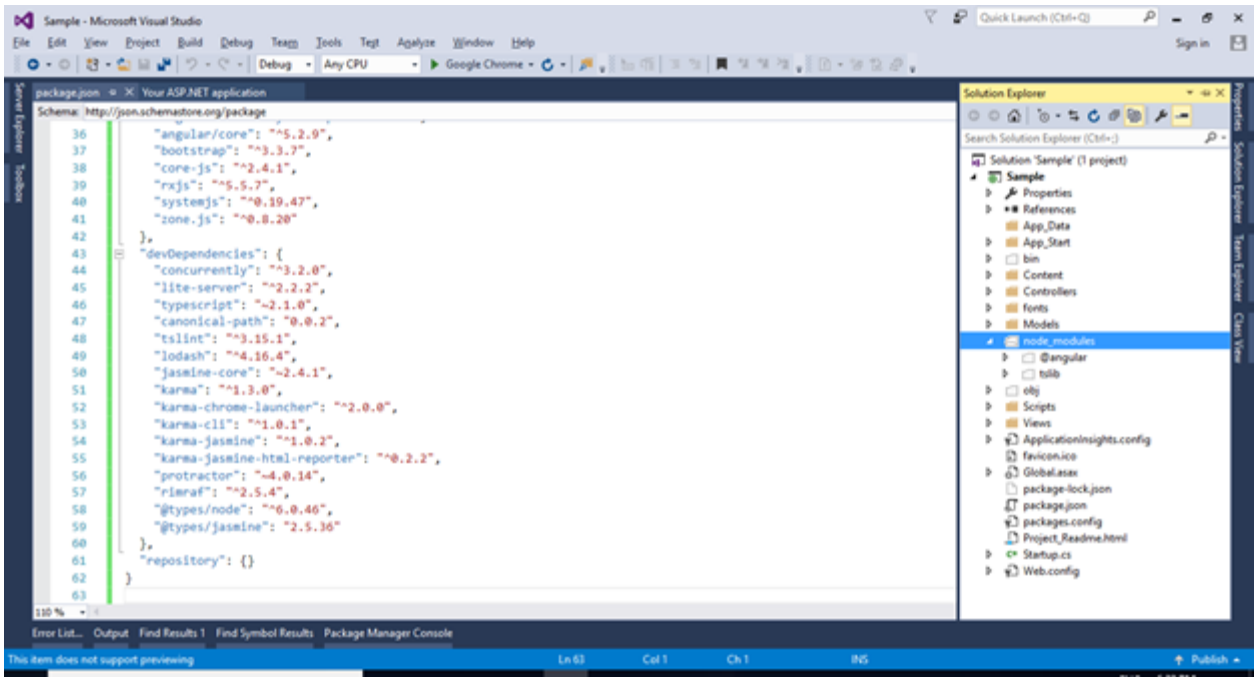
```
31.          "protractor": "protractor protractor.config.js",
32.
33.          "pretest": "npm run build",
34.
35.          "test": "concurrently \"npm run build:watch\" \"karma
   start karma.conf.js\"",
36.
37.          "pretest:once": "npm run build",
38.
39.          "test:once": "karma start karma.conf.js --single-run",
40.
41.          "lint": "tslint ./src/**/*.ts -t verbose"
42.
43.        },
44.
45.        "keywords": [],
46.
47.        "author": "",
48.
49.        "license": "MIT",
50.
51.        "dependencies": {
52.
53.          "@angular/common": "~4.3.4",
54.
55.          "@angular/compiler": "~4.3.4",
56.
57.          "@angular/core": "~4.3.4",
58.
59.          "@angular/forms": "~4.3.4",
60.
61.          "@angular/http": "~4.3.4",
62.
63.          "@angular/platform-browser": "~4.3.4",
64.
65.          "@angular/platform-browser-dynamic": "~4.3.4",
66.
67.          "@angular/router": "~4.3.4",
68.
69.          "angular-in-memory-web-api": "~0.3.0",
70.
71.          "systemjs": "0.19.40",
72.
73.          "core-js": "^2.4.1",
74.
75.          "rxjs": "5.0.1",
76.
77.          "zone.js": "^0.8.4"
78.
79.        },
80.
81.        "devDependencies": {
82.
83.          "concurrently": "^3.2.0",
84.
85.          "lite-server": "^2.2.2",
86.
87.          "typescript": "~2.1.0",
88.
89.          "canonical-path": "0.0.2",
90.
```

```
91.           "tslint": "^3.15.1",
92.
93.           "lodash": "^4.16.4",
94.
95.           "jasmine-core": "~2.4.1",
96.
97.           "karma": "^1.3.0",
98.
99.           "karma-chrome-launcher": "^2.0.0",
100.
101.          "karma-cli": "^1.0.1",
102.
103.          "karma-jasmine": "^1.0.2",
104.
105.          "karma-jasmine-html-reporter": "^0.2.2",
106.
107.          "protractor": "~4.0.14",
108.
109.          "rimraf": "^2.5.4",
110.
111.          "@types/node": "^6.0.46",
112.
113.          "@types/jasmine": "2.5.36"
114.
115.       },
116.
117.       "repository": {}
118.
119.    }
```

Después de agregar el código, haga clic derecho en el archivo y elija la opción “Restore Package”



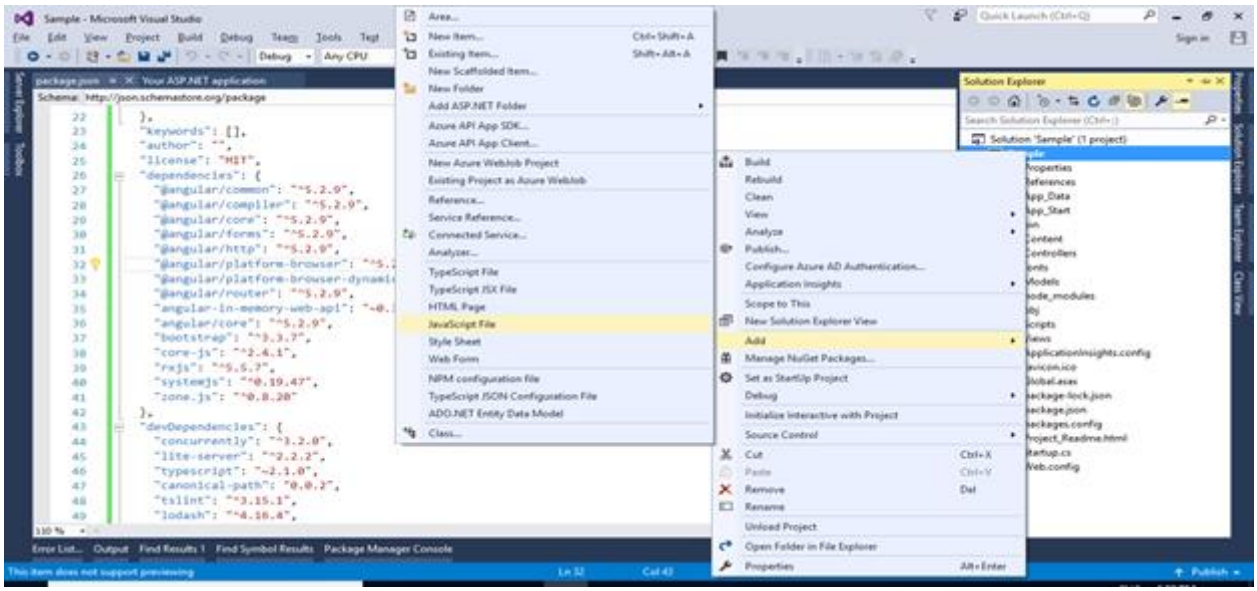
□□□• Después de actualizar, verá la carpeta "node_modules" en el proyecto



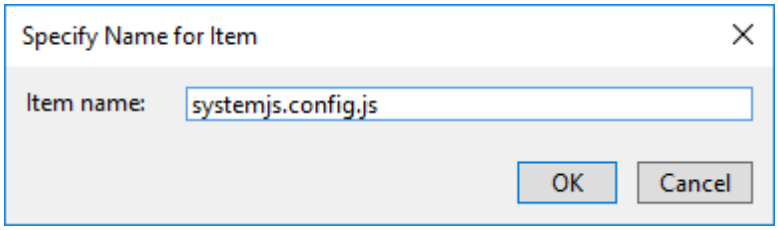
Paso 3: Agregue el archivo de script java de configuración del sistema.

□□□• Agregue el archivo systemjs.config.js para especificar la ruta de referencia a los componentes

•□Haga clic derecho sobre el proyecto, Añadir ↪·archivo JavaScript



- Dé el nombre como "systemjs.config.js" y haga clic en Aceptar.



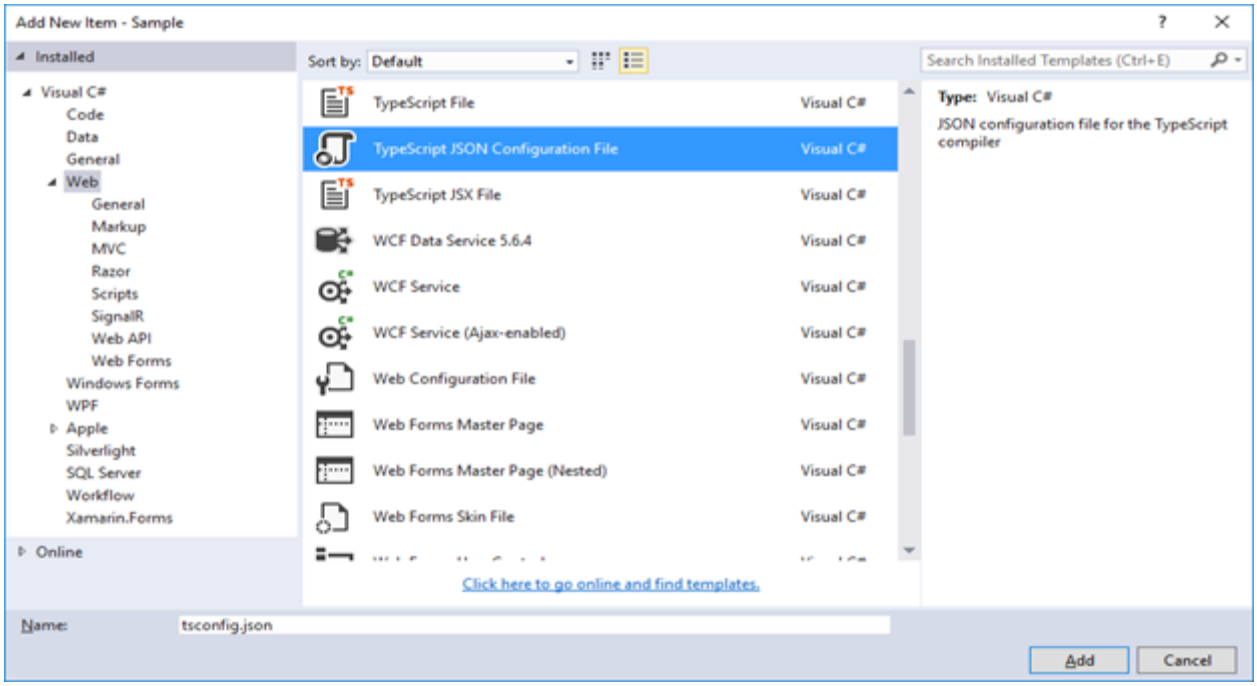
- Pegue el fragmento de código a continuación en el archivo systemjs.config.js creado.

```
System.config({

    paths: {

      // paths serve as alias

      'npm:': 'node_modules/'

    },

        // map tells the System loader where to look for things

        map: {

          // our app is within the app folder

          'app': 'app',

          // angular bundles

          '@angular/core': 'npm:@angular/core/bundles/core.umd.
  js',

          '@angular/common': 'npm:@angular/common/bundles/commo
  n.umd.js',

          '@angular/compiler': 'npm:@angular/compiler/bundles/c
  ompiler.umd.js',

          '@angular/platform-browser': 'npm:@angular/platform-
  browser/bundles/platform-browser.umd.js',

          '@angular/platform-browser-
  dynamic': 'npm:@angular/platform-browser-
  dynamic/bundles/platform-browser-dynamic.umd.js',

          '@angular/http': 'npm:@angular/http/bundles/http.umd.
  js',

          '@angular/router': 'npm:@angular/router/bundles/route
  r.umd.js',

          '@angular/forms': 'npm:@angular/forms/bundles/forms.u
  md.js',

```

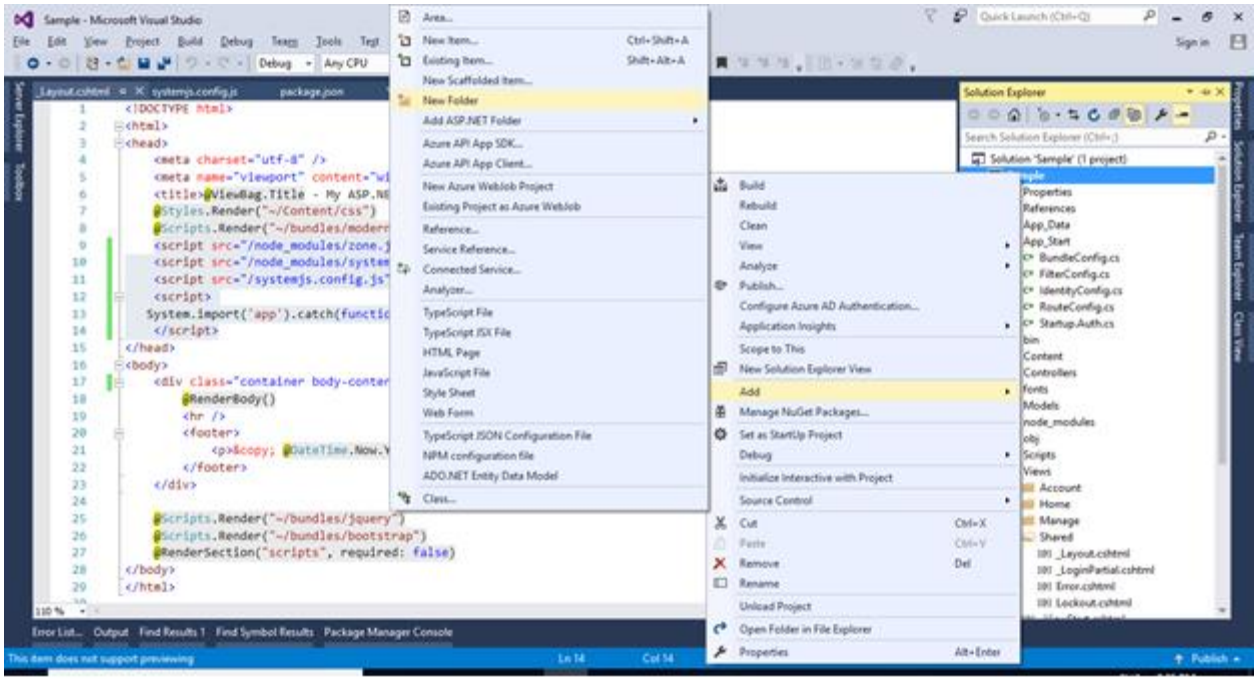
```
            // other libraries

            'rxjs':                      'npm:rxjs',

            'angular-in-memory-web-api': 'npm:angular-in-memory-
   web-api/bundles/in-memory-web-api.umd.js'

          },

          // packages tells the System loader how to load when no
   filename and/or no extension

          packages: {

            app: {

              defaultExtension: 'js',

              meta: {

                './*.js': {

                  loader: 'systemjs-angular-loader.js'

                }

              }

            },

            rxjs: {

              defaultExtension: 'js'

            }

          }

        });
```

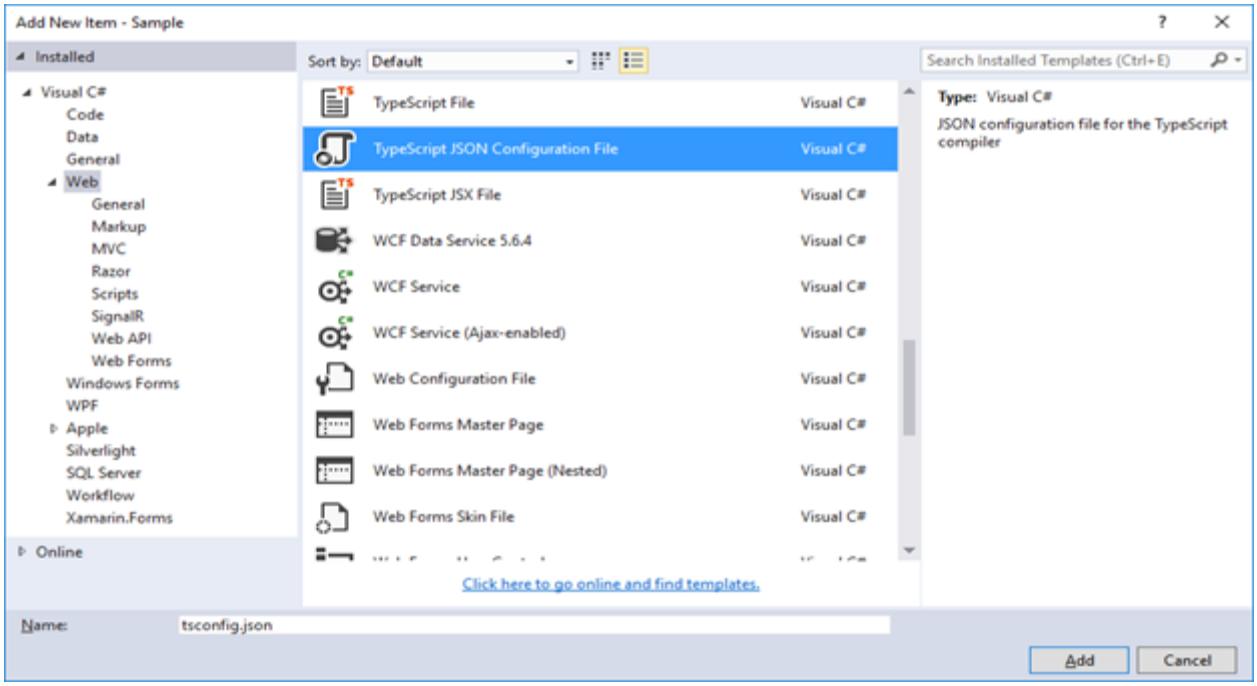
Paso 4: Agregue el archivo de configuración de typeScript.

- Agregue el archivo tsconfig.json para especificar las opciones del compilador para compilar



□ □ □• Elija este archivo de configuración JSON para typeScript y pegue el fragmento de código a continuación en el archivo creado tsconfig.json.






```
{
  "compilerOptions": {
    "target": "es5",
    "module": "commonjs",
    "moduleResolution": "node",
    "sourceMap": true,
    "emitDecoratorMetadata": true,
    "experimentalDecorators": true,
    "lib": [ "es2015", "dom" ],
          "noImplicitAny": true,
          "suppressImplicitAnyIndexErrors": true
        }
    }
```

Abra el archivo "RouteConfig.cs" y agregue el siguiente código para permitir URL que acepte entradas de Angular,

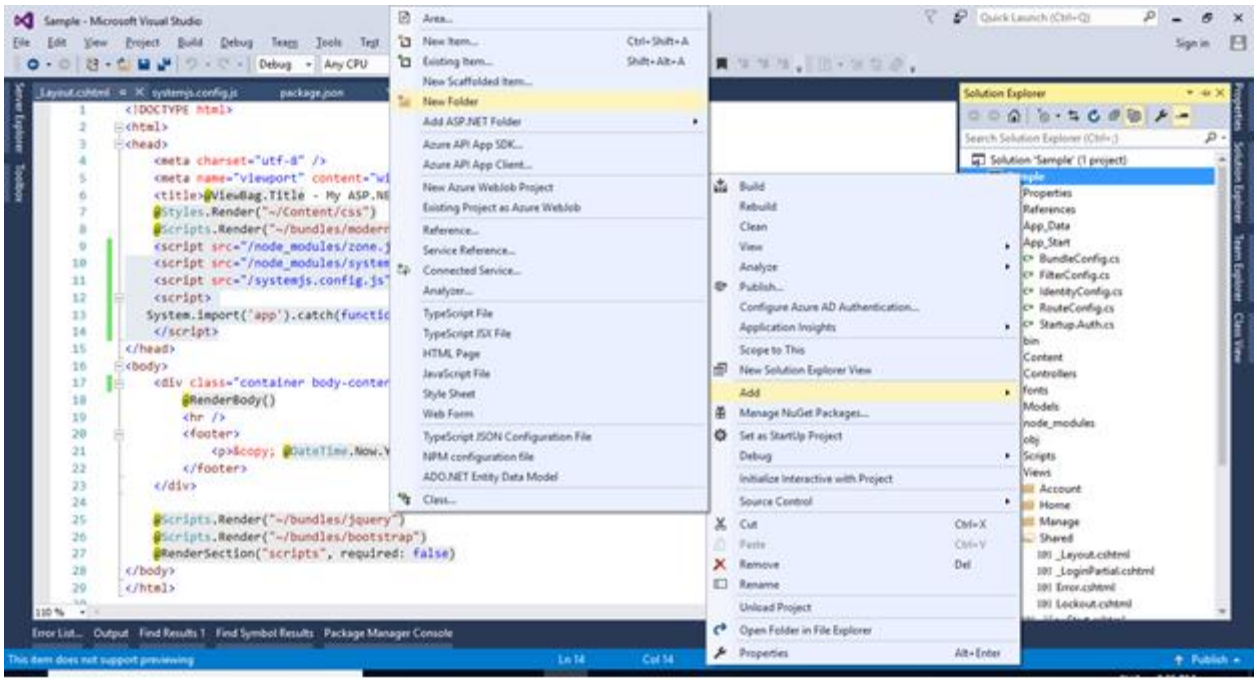
```
routes.MapRoute(

                name: "angular",

                url: "{*anything}",

                defaults: new { controller = "Home", action = "I
ndex" } // The view that bootstraps Angular 2

            );
```

Paso 6: Agregar referencia angular

- Agregue referencia angular incluyendo el fragmento a continuación en el archivo de diseño.

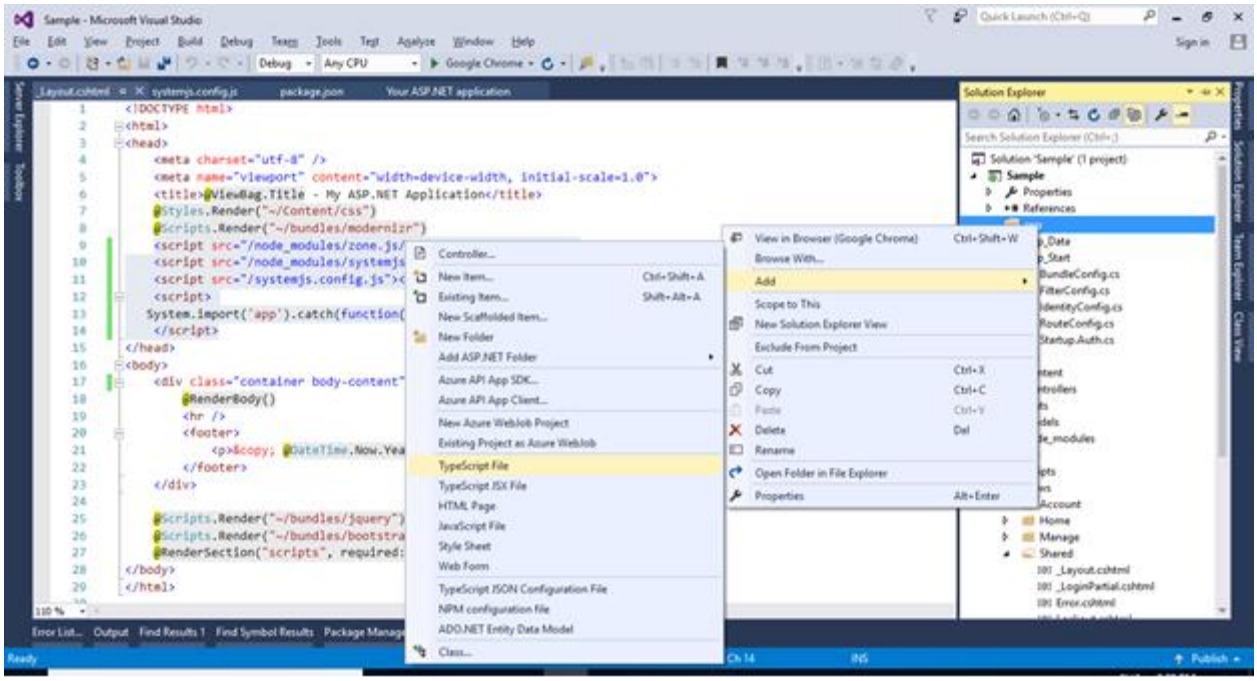
```
10.      <script src="/node_modules/zone.js/dist/zone.js"></script>
11.
12.          <script
   src="/node_modules/systemjs/dist/system.src.js"></script>
13.
14.          <script src="/systemjs.config.js"></script>
15.
16.          <script>
17.
18.         System.import('app').catch(function(err){ console.error(
   err); });
19.
20.          </script>
```

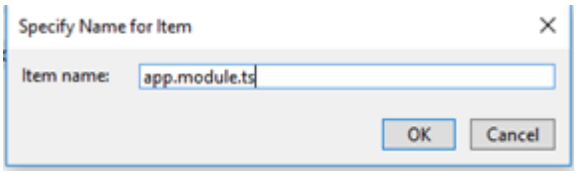
Añade la carpeta APP:



Paso 7: configurar la estructura angular

- Crear archivo de módulo de aplicación angular "app.module.ts"



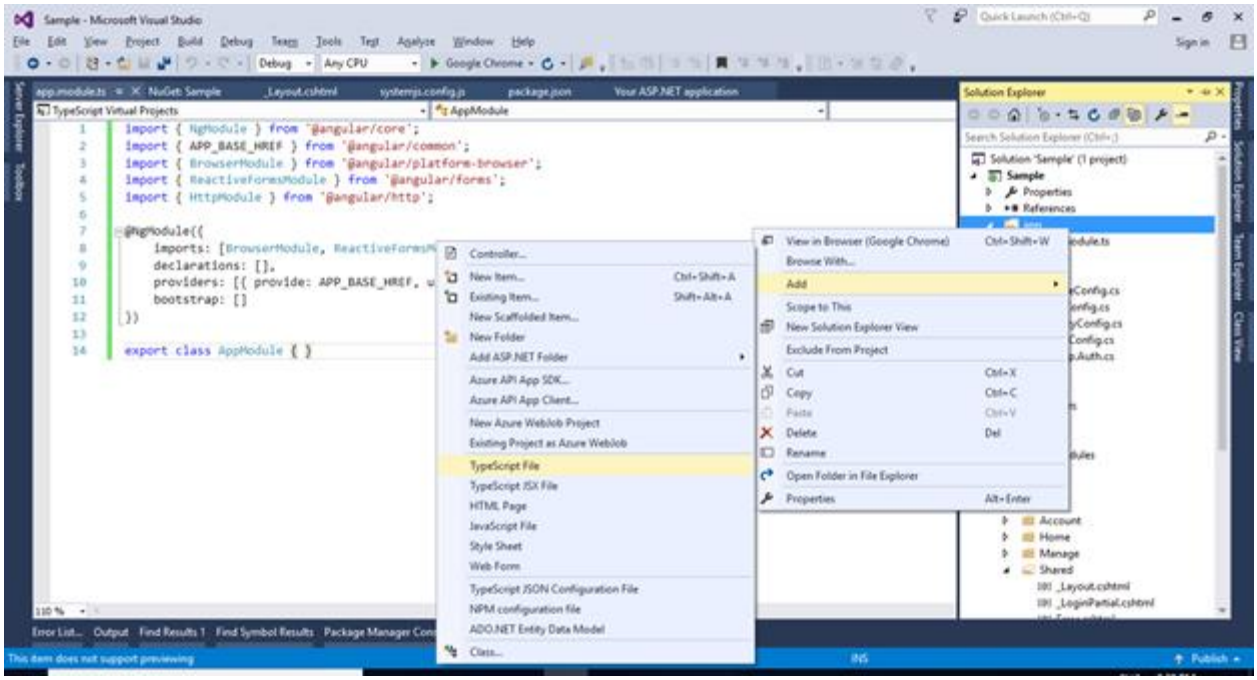


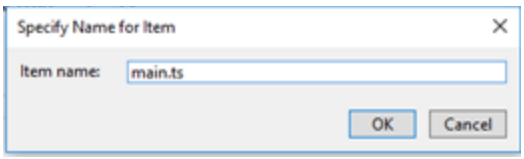
- Pegue el fragmento de código a continuación en el módulo de la aplicación, app.nodule.ts:

```
1. import { NgModule } from '@angular/core';
2.
3. import { APP_BASE_HREF } from '@angular/common';
4.
5. import { BrowserModule } from '@angular/platform-browser';
6.
7. import { ReactiveFormsModule } from '@angular/forms';
8.
9. import { HttpModule } from '@angular/http';
10.
11.       import { AppComponent } from './app.component';
12.
13.       import { routing } from './app.routing';
14.
15.       import { HomeComponent } from './Component/home.component';
16.
17.       import { AboutComponent } from './Component/about.component
    ';
18.
19.       @NgModule({
20.
21.           imports: [BrowserModule, ReactiveFormsModule, HttpModul
    e, routing],
```

```
22.
23.          declarations: [AppComponent, HomeComponent, AboutCompon
   ent],
24.
25.          providers: [{ provide: APP_BASE_HREF, useValue: '/' }],
26.
27.          bootstrap: [AppComponent]
28.
29.      })
30.
31.      export class AppModule { }
```

Crear el archivo principal de angular main.ts





Pega el siguiente código en el fichero:

```
1. import { platformBrowserDynamic } from '@angular/platform-
   browser-dynamic';
2.
3. import { AppModule } from './app.module';
4.
5. platformBrowserDynamic().bootstrapModule(AppModule);
6.
```

```
7.
8.  ·         Create app component file “app.component.ts” and paste
    the below code snippet,
9.
10.       import { Component } from "@angular/core"
11.
12.       @Component({
13.
14.           selector: "user-app",
15.
16.           template: `
17.
18.                       <div>
19.
20.                          <nav class='navbar navbar-inverse'>
21.
22.                               <div class='container-fluid'>
23.
24.                                 <ul class='nav navbar-nav'>
25.
26.                                   <li><a [routerLink]="['home']">H
    ome</a></li>
27.
28.                                   <li><a [routerLink]="['about']">
    About</a></li>
29.
30.                               </ul>
31.
32.                               </div>
33.
34.                          </nav>
35.
36.                       <div class='container'>
37.
38.                         <router-outlet></router-outlet>
39.
40.                       </div>
41.
42.                     </div>
43.
44.       `
45.
46.       })
47.
48.       export class AppComponent {
49.
50.
51.       }
```

Cree el archivo de componente de aplicación "app.routing.ts" y actualice el enrutamiento para los módulos creados.

- Crear los componentes home y about

- Cree el archivo de componente de la aplicación "home.component.ts" y pegue el fragmento de código a continuación,

```
1. import { Component } from "@angular/core";
2.
3. @Component({
4.
5.     template: `<p>This is home page.</p>`
6.
7. })
8.
9. export class HomeComponent {
10.
11.
12.      }
13.
14.
15.      ·         Create app component file “about.component.ts”
   and paste the below code snippet,
16.
17.      import { Component } from "@angular/core";
18.
19.      @Component({
20.
21.          template: `<p>This is about page.</p>`
22.
23.      })
24.
25.      export class AboutComponent {
26.
27.
28.      }
```

Paso 8: Agregar referencia de código en el archivo index.cshtml.

•□ Elimine el código existente y pegue el fragmento de código a continuación en el archivo index.cshtml

```
1. @{
2.
3.     ViewBag.Title = "Index";
4.
5. }
6.
7. <body>
8.
9.     <user-app> This is normal ASP.Net MVC</user-app>
10.
11.
12.      </body>
```

Paso 9: compilar y ejecutar la aplicación

•□ Cargue la salida angular ingresando la URL " http: // localhost: portnumber / home "

- Haga clic en el enlace sobre para ver la navegación,

•□Cargue la salida MVC ingresando la url " http: // localhost: portnumber / home / index "