**Lesson02 Angular 4 File Structure**

**Notes:-**

**1-Angular 4 are enhanced product from the Angular 2 that is initialized in 2015 and is built in 2015 from scratch**

**2-the basic element is Component**

**3-the benefit of the Angular 4 is the small size and high execution**

**Steps:-**

**1-on the Visual Studio code 🡪 Extensions tab 🡪**

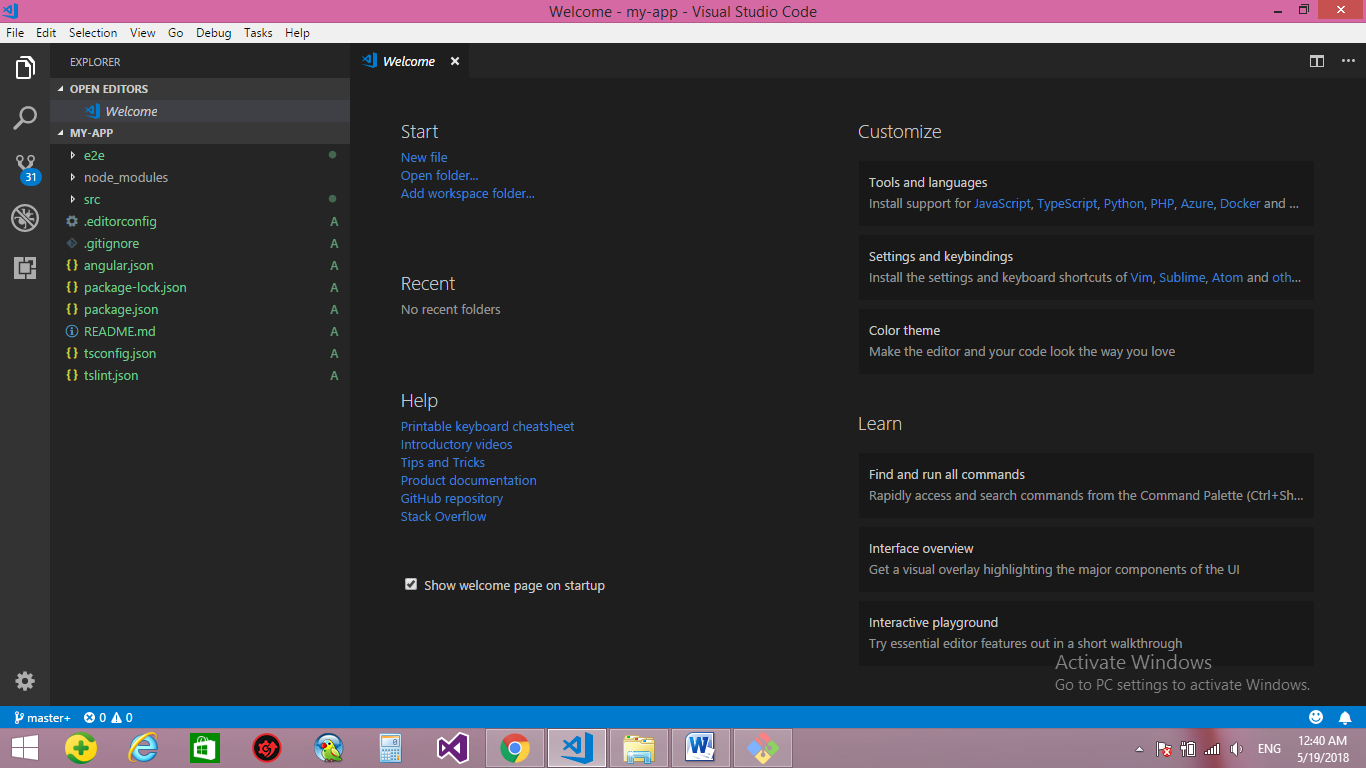
**Select the Angular v4 Typescript Snippet**

**(This means that it’s automatically complete the code you put it)**

# Select the Angular 2, 4 Typescript, HTML Snippets for VS Code

# (to make autocomplete of html element with creation in angular 4)

**2-on the Visual Studio code 🡪 file 🡪 Open Folder 🡪 Select the project we created before 🡪 Click Ok**



**Package.json:- it’s the manifest of the application that carry the application name and the application version**

{"name": "my-app",

"version": "0.0.0",

//we see that the script make connect between the action performed on the application and the commands associated to it

"scripts": {

"ng": "ng",

"start": "ng serve",

"build": "ng build",

"test": "ng test",

"lint": "ng lint",

"e2e": "ng e2e"

},

"private": true,

"dependencies": {

"@angular/animations": "^6.0.2",

"@angular/common": "^6.0.2",

"@angular/compiler": "^6.0.2",

"@angular/core": "^6.0.2",

"@angular/forms": "^6.0.2",

"@angular/http": "^6.0.2",

"@angular/platform-browser": "^6.0.2",

"@angular/platform-browser-dynamic": "^6.0.2",

"@angular/router": "^6.0.2",

"core-js": "^2.5.4",

"rxjs": "^6.0.0",

"zone.js": "^0.8.26"

},

"devDependencies": {

"@angular/compiler-cli": "^6.0.2",

"@angular-devkit/build-angular": "~0.6.3",

//we see that this is the version of the Typescript that build your application

"typescript": "~2.7.2",

"@angular/cli": "~6.0.3",

"@angular/language-service": "^6.0.2",

"@types/jasmine": "~2.8.6",

"@types/jasminewd2": "~2.0.3",

"@types/node": "~8.9.4",

"codelyzer": "~4.2.1",

"jasmine-core": "~2.99.1",

"jasmine-spec-reporter": "~4.2.1",

"karma": "~1.7.1",

"karma-chrome-launcher": "~2.2.0",

"karma-coverage-istanbul-reporter": "~1.4.2",

"karma-jasmine": "~1.1.1",

"karma-jasmine-html-reporter": "^0.2.2",

"protractor": "~5.3.0",

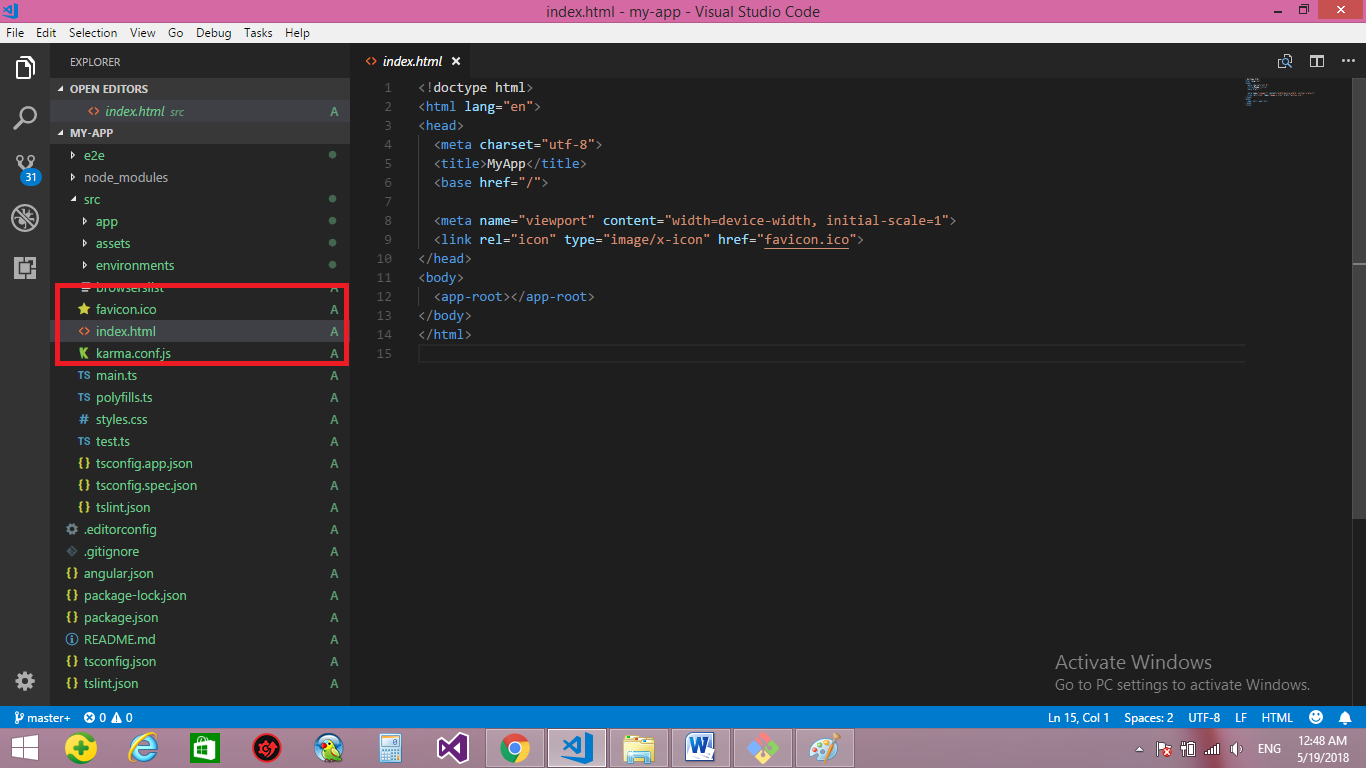
"ts-node": "~5.0.1",

"tslint": "~5.9.1"

}}

**3-on the src 🡪 Index.cshtml 🡪 we see that that there are the default page and we see that the below code represent the icon on the tab**

<link rel="icon" type="image/x-icon" href="favicon.ico">

****

**4-on the src 🡪 main.ts 🡪 we see the following code**

//we import the production library angular/core

import { enableProdMode } from '@angular/core';

//we import the library responsible to run the angular to the browser

import { platformBrowserDynamic } from '@angular/platform-browser-dynamic';

//we enable the module structure on the project , to make our project more extensively and dynamically and testable

import { AppModule } from './app/app.module';

//we import the environment on the applicaion

import { environment } from './environments/environment';

//if the environment is production enable the production mode

if (environment.production) {

enableProdMode();

}

//This is the basic module that run the application to it

platformBrowserDynamic().bootstrapModule(AppModule)

.catch(err => console.log(err));

**5-on the tsconfig.app.json we see that**

{

//This is means that the file inherit from the tsconfig.json

"extends": "../tsconfig.json",

"compilerOptions": {

//This means that the output go to the specific path

"outDir": "../out-tsc/app",

//This means that the TypeScript is TypeScript 2015 , 6

"module": "es2015",

"types": []

},

"exclude": [

"src/test.ts",

"\*\*/\*.spec.ts"

]

}

//any file end with .ts must be compiled from type script to java script to allow the browser to read the JavaScript

**6-the most component we focus is the src 🡪 app (we see multiple files)**

**1-app.component.ts :- this file contain the selector that map between the Index.html that search to the html page and the target page map on it**

import { Component } from '@angular/core';

@Component({

selector: 'app-root',

templateUrl: './app.component.html',

styleUrls: ['./app.component.css']

})

export class AppComponent {

title = 'app';

}

<!doctype html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>MyApp</title>

<base href="/">

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="icon" type="image/x-icon" href="favicon.ico">

</head>

<body>

//we see that when browsing the page , it will searching for the selector assigning to it and put the html page and css file and title on it

<app-root></app-root>

</body>

</html>