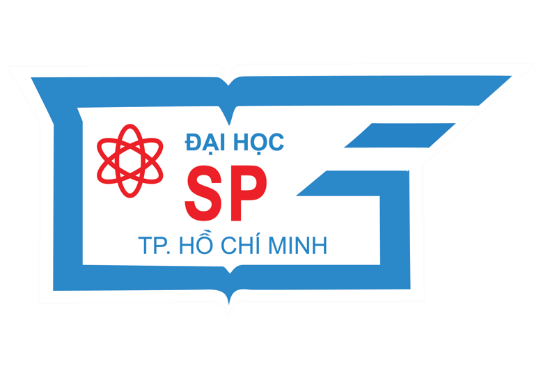
**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM THÀNH PHỐ HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**--🕮--**



**BÁO CÁO ĐỀ TÀI GIỮA KÌ**

**ANGULAR 8 VÀ ASP.NET CORE**

**Giảng viên hướng dẫn:** Ths. Lương Trần Hy Hiển

**Nhóm sinh viên thực hiện:**

Lý Thành Nam 43.01.104.111

Phan Minh Tuấn 43.01.104.200

Đào Phú Nghĩa 43.01.104.115

Trần Thu Thủy 43.01.104.177

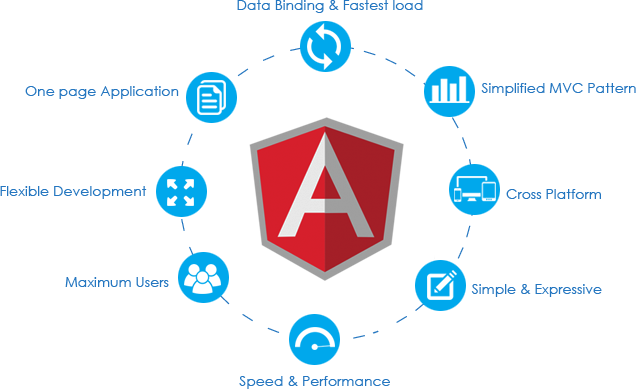
Võ Yến Nhi 43.01.104.119

Quách Trí Tân 43.01.104.159

Tháng 12/2019

1. **Giới thiệu tổng quan:**
2. **Angular là gì?**

Angular là một Framework được tạo ra để xây dựng các ứng dụng web động (dynamic web app), nó thường được sử dụng để tạo ra các ứng dụng một trang ( Single Page Application - SPA). Và hoạt động dựa trên việc mở rộng thêm các thuộc tính (attribute) cho các thẻ của HTML (Các thuộc tính theo quy tắc của Angular). Angular là một nền tảng miễn phí và được hàng ngàn lập trình viên trên thế giới ưa chuộng.



1. **Sự phát triển:**

Thực tế là Angular có 2 phiên bản hoàn toàn khác nhau, phiên bản 1 và phiên bản khác 1.

**Version 1:**

Phiên bản 1 có tên chính thức là AngularJS, hiện tại nó vẫn đang được phát triển. Phiên bản Angular1 thực sự dựa trên Javascript, các lập trình viên sẽ viết mã theo ngôn ngữ Javascript.

**Version 2, 3, 4, ..**

Bắt đầu từ phiên bản 2 nó có tên chính thức là Angular. Angular phiên bản 2 trở lên có tư duy thay đổi hoàn toàn so với Angular1. Tới nay, Angular đã phát triển lên version 8, được phát hành cuối tháng 5/2019 với những cải tiến giúp hiệu năng tốt hơn, code dễ và nhanh hơn.

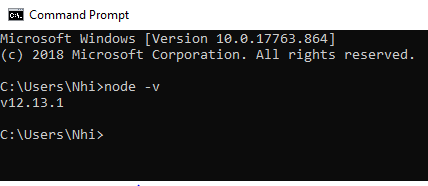
1. **Cài đặt môi trường:**

Để tạo một dự án Angular, chúng ta có thể sử dụng công cụ Angular CLI. Nó là một công cụ vô cùng hữu hiệu giúp chúng ta có thể xây dựng các ứng dụng Angular một cách nhanh hơn và hiệu quả hơn. Ở đây, để cài đặt Angular, cần một số công cụ:

1. Cài đặt Nodejs:

Angular ở phiên bản mới nhất yêu cầu Nodejs version 10.9.0 trở lên.

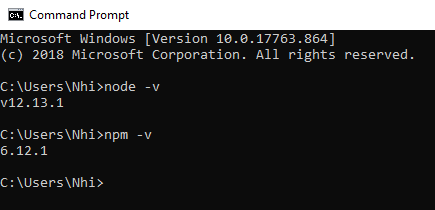
* Nếu máy tính đã cài đặt Nodejs, để kiểm tra version Nodejs trên máy tính, chạy câu lệnh node –v trong màn hình Command line.



* Nếu máy tính chưa cài đặt Nodejs, truy cập vào đường link download nodejs.org để tải về.

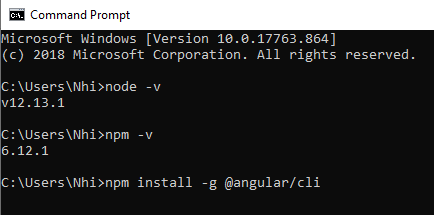
1. Cài đặt npm:

Cài được NodeJS thành công, máy bạn sẽ tự có npm (tức npm đã được cài đặt). Để kiểm tra phiên bản npm, gõ npm –v

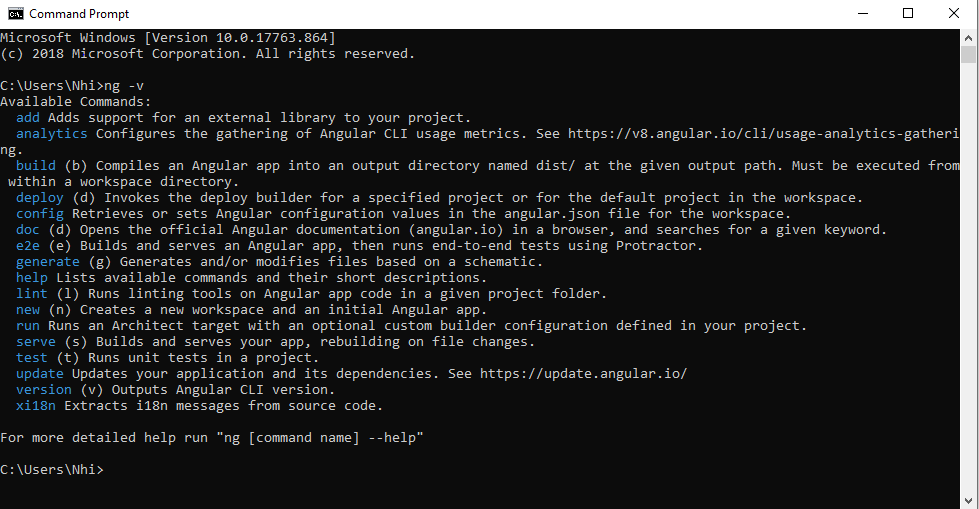


1. Cài đặt Angular CLI:

Để cài đặt Angular CLI thì bạn gõ npm install -g @angular/cli trong màn hình Command line. Chờ khoảng vài phút, tiến trình cài đặt hoàn tất.

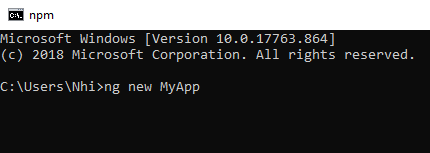


Để kiểm tra, bạn gõ ng -v trên cửa sổ command line. Nếu hiển thị như bên dưới là bạn đã cài đặt thành công Angular CLI.

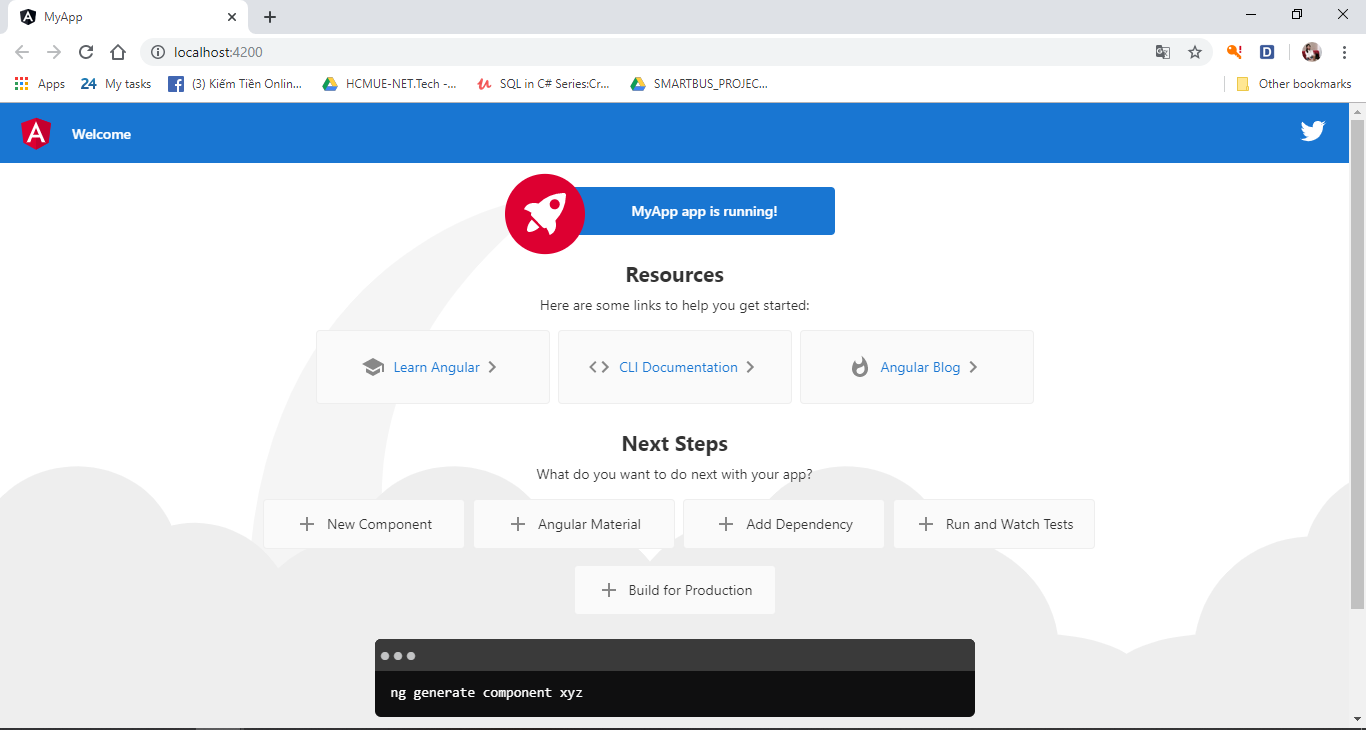


1. Tạo dự án Angular:

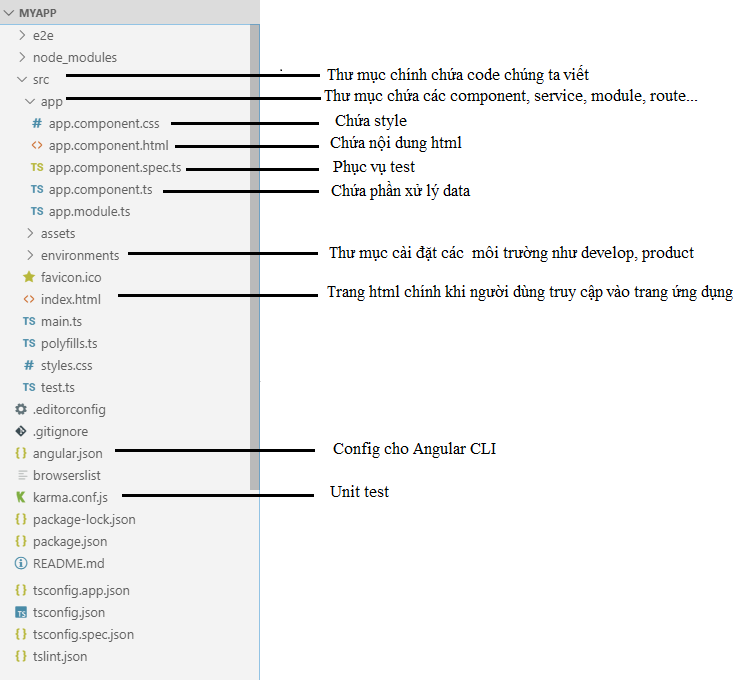
* Sau khi cài đặt xong Angular CLI thì việc tạo một dự án Angular có thể được thực hiện một cách đơn giản thông qua việc chạy câu lệnh sau: ng new [app\_name]



* Sau khi chạy xong đoạn lệnh, sẽ xuất hiện thư mục tên MyApp trong thư mục tương ứng với đường dẫn thư mục trong Command line. Để chạy chương trình, ta chạy lệnh ng serve –o Kết quả chương trình sẽ chạy trên cửa sổ trình duyệt với giao diện như hình phía dưới.



1. **Tìm hiểu Angular:**
2. **Cấu trúc thư mục:**

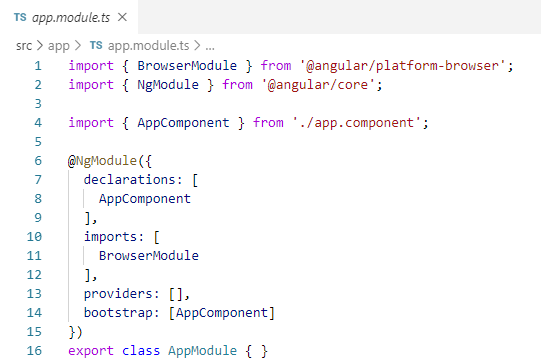


Để có thể xây dựng một app Angular, chúng ta cần có một số kiến thức về ES6, Typescript, Angular CLI, Components, Dependency Injection, Directives, Form, Pipes, HTTP, Service, Routing.

1. **Components:**

Phần lớn quá trình phát triển ứng dụng Angular là sử dụng components. Components (hay thành phần) là những class cơ bản tương tác với file html. Có một component cơ bản luôn được tạo ra khi tạo mới project là AppComponent.

1. Thành phần cha AppComponent:



Trong file app.module.ts , 3 dòng đầu tiên có tác dụng import các module và component cần thiết: BrowserModule, NgModule, AppComponent.

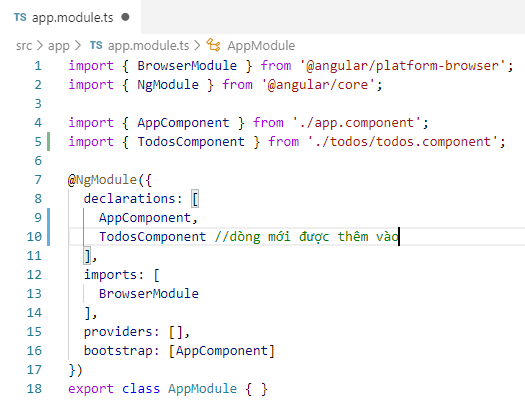
BrowserModule và NgModule là 2 module căn bản app nào cũng có. AppComponent là component được khai báo trong file app.component.ts.

AppComponent là thành phần cha của ứng dụng. Tất cả các thành phần mới tạo ra sau này đều là thành phần con của AppComponent.

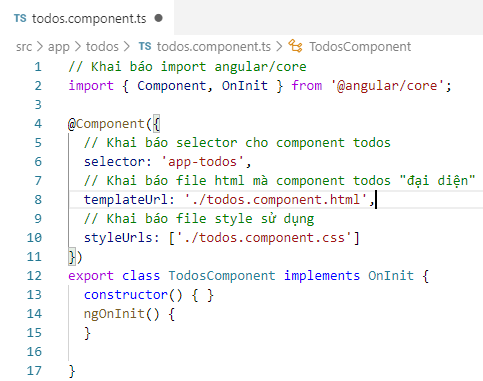
Để tạo component mới, ta sử dụng lệnh: ng g component <ten-component>

Sau khi chạy lệnh, một thư mục <ten-component> sẽ xuất hiện trong thư mục src/app có cấu trúc tương tự như thư mục app.

Ngoài việc tạo ra mới 4 file này, angular CLI cũng giúp chúng ta khai báo new component này cho project trong file app.module.ts



1. Cấu trúc của component:



Ở đây khai báo 1 class gọi là TodosComponent. Class này implements class OnInit. Class OnInit có constructor là ngOnInit(), nên trong TodosComponent cũng sẽ override lại constructor ngOnInit() này. Constructor ngOnInit() sẽ được gọi mặc định khi class được gọi đến.

Ta biết được luồng hoạt động của AppComponent: file main.ts được chạy đầu tiên. Trong file main.ts khai báo AppModule. AppModule khai báo sử dụng AppComponent. Trong App Component có selector <app-root>, <app-root> được gọi trong file index.html và cuối cùng file index.html được hiển thị lên trình duyệt.

Cách hoạt động của component khác cũng tương tự như App Component. Trong ví dụ trên, TodosComponent khai báo selector là <app-todos>. Như vậy thẻ <app-todos> sẽ "đại diện" cho TodosComponent, khai báo <app-todos> ở đâu thì nội dung file html của TodosComponent sẽ được hiển thị ở chỗ đó, giống như khi khai báo <app-root></app-root> ở đâu thì nội dung file app.component.html được hiển thị ở đó.

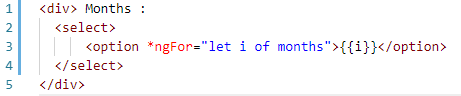
1. **Data Binding:**

Data binding là cơ chế lấy dữ liệu từ trong component và đưa ra view. Data Binding đã có từ AngularJS, Angular 2 và giờ cũng có ở Angular 4. Để bind dữ liệu ra view, ta sử dụng cặp ngoặc nhọn {{ten\_bien}}. ten\_bien chính là tên biến bạn đặt phía bên trong file component.ts

Ở đây, ta tìm hiểu cách data biding với ngFor. Trong file appcomponent.ts, ta khai báo một biến mảng chứa danh sách các tháng:



Tiếp đó, ta bind dữ liệu ra view:



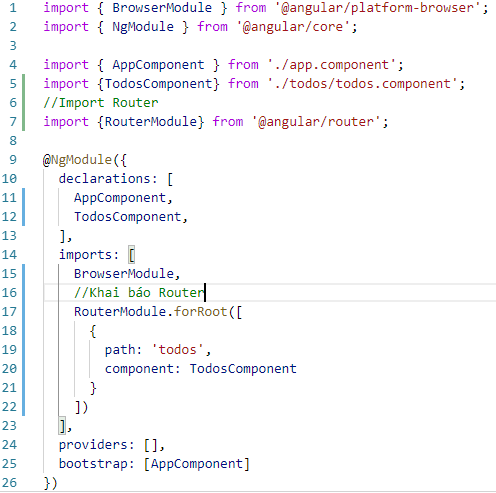
months là một mảng chứa các tháng. Câu lệnh let i of months đưa lần lượt dữ liệu các phần tử của mảng months vào biến i, nghĩa là i sẽ lần lượt chứa các giá trị Tháng 1, Tháng 2,..., tháng 12. Biến i nằm trong cặp ngoặc {{i}} sẽ bind dữ liệu của i ra view.

1. **Routing:**

Routing là cơ chế chuyển giữa các trang và thay đổi dữ liệu ở thanh điều hướng của trình duyệt. Trong bài này, chúng ta sẽ tìm hiểu về routing trong Angular 4: các thành phần, ý nghĩa của chúng và sử dụng routing để hiển thị NewCmpComponent.

1. Khai báo Route:

Trong file app.module.ts, chúng ta khai báo sử dụng RouterModule

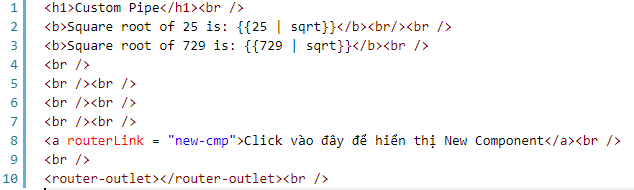


**RouterModule.forRoot()** nhận vào một mảng là các giá trị của router. Mỗi phần tử bao gồm path và component. Trong đó:

* **path**: đường dẫn
* **component**: component mà đường dẫn đó dùng tới.

1. Sử dụng Router:

Trong file **app.component.html**, ta chỉnh sửa như sau:



Chúng ta vừa tạo ra 1 thẻ <a> với thuộc tính routerLink="new-cmp". Đây chính là path mà ta đã định nghĩa ở file **app.module.ts**. Cặp thẻ <router-outlet></router-outlet> đặt ở đâu thì nội dung của component sẽ hiển thị ở đó.

1. **Pipe:**

Pipe được sử dụng để chuyển đổi dữ liệu. Cú pháp sử dụng pipe:

{{du\_lieu | pipe}}

* **du\_lieu**: là dữ liệu cần chuyển đổi
* **pipe**: là pipe mà bạn sử dụng.

Angular xây dựng sẵn 1 số built-in pipe thường dùng như bảng dưới.

|  |  |
| --- | --- |
| **Directive** | **Ghi chú** |
| Lowercase | Chuyển chuỗi thành kí tự thường |
| Uppercase | Chuyển chuỗi thành kí tự hoa |
| Date | Chuyển dữ liệu thành định dạng ngày tháng |
| Currency | Chuyển dữ liệu thành định dạng tiền tệ |
| Json | Chuyển dữ liệu về dạng JSON |
| Percent | Chuyển dữ liệu về dạng % |
| Decimal | Chuyển dữ liệu về dạng thấp phân |
| slice | Cắt dữ liệu của mảng. Giống như Array.prototype.slice() |

1. **Typescript:**
2. Typescript là gì?

* TypeScript là một dự án mã nguồn mở được phát triển bởi Microsoft. Nó có thể được coi là một phiên bản nâng cao của Javascript bởi việc bổ sung tùy chọn kiểu tĩnh và lớp hướng đối tượng mà điều này không có ở Javascript.
* TypeScript có thể sử dụng để phát triển các ứng dụng chạy ở client-side (Angular) và server-side (NodeJS).
* TypeScript sử dụng tất cả các tính năng của của ECMAScript 2015 (ES6) như classes, modules.

1. Cài đặt Typescript:

* Để cài đặt, chúng ta chạy lệnh npm install –g typescript
* Typescript có đuôi mở rộng là .ts
* Để biên dịch một file Typescript sang javascript ta chạy lệnh:

tsc tên\_file –watch

Sau khi chạy lệnh, 1 file .js sẽ tự động sinh ra để chứa các code của file .ts

1. Kiểu dữ liệu và khai báo biến:

TypeScript hỗ trợ tất cả các kiểu dữ liệu của JavaScript và bổ sung thêm một số kiểu dữ liệu hiện đại để giúp cho lập trình viên làm việc dễ dàng hơn.

|  |  |  |
| --- | --- | --- |
|  | **Javascript** | **Typescript** |
| Kiểu dữ liệu cơ bản | Number, string, Boolean, array… | …, enum, tuple, any, void, … |
| Cú pháp khai báo | Var test = 123; | Var test : string = “123” |

Trong đó:

* Tuple: cho phép bạn khai báo mảng với các giá trị có kiểu dữ liệu mà bạn đã biết. Như ví dụ dưới đây

// Declare a tuple type

let x: [string, number];

// Initialize it

x = ["hello", 10];

* Enum: Cũng giống như C#, enum là một cách để đặt những cái tên thân thiện hơn với bộ giá trị số.

enum Color {Red, Green, Blue};

let c: Color = Color.Green;

* Any: Đôi khi chúng ta cần một kiểu dữ liệu mà chúng ta không biết chắc chắn kiểu dữ liệu của nó, những giá trị này là nội dung động như từ một thư viện khác hoặc người dùng nhập. Để làm như vậy, chúng ta sử dụng any.

let notSure: any = 4;

notSure = "maybe a string instead";

notSure = false; // okay, definitely a boolean

1. Function:

* Trong typescript các hàm đều trả về một kết quả, kết quả đó sẽ thuộc về một kiểu dữ liệu nào đó (vd: number, string, void, …).
* Ta phải chỉ định luôn kiểu dữ liệu trả về cho hàm ngay từ đầu.

function sum (x: number, y: number) : number {

return x + y;

}

Hoặc là:

var z = function(x: number, y: string) : string {

return `Chao ${y}, nam nay ban ${x} tuoi phai khong?`;

}

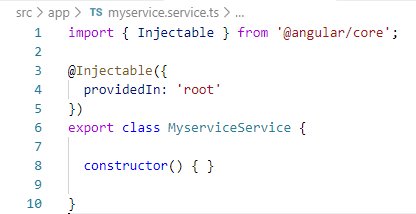
1. **Service:**

Có một số tình huống mà chúng ta cần sử dụng một số đoạn code ở nhiều trang khác nhau ví dụ như chia sẻ dữ liệu giữa các component với nhau. Để làm được điều này, chúng ta có thể sử dụng service. Với service, chúng ta có thể truy cập các phương thức, thuộc tính của các component trong cả project.

1. Tạo service:

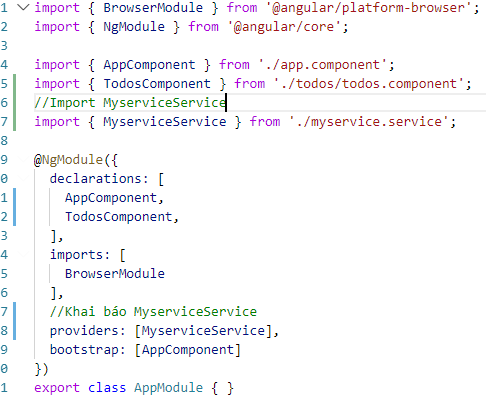
Để tạo service, ta chạy câu lệnh sau trên màn hình command line

ng g service myservice. Sau khi chạy xong, ta có thêm 2 file nữa trong project là **myservice.service.ts** và **myservice.service.spect.ts**

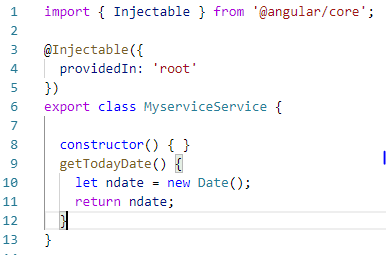


1. Khai báo service:

Để sử dụng myservice service, chúng ta cần khai báo trong **app.module.ts**



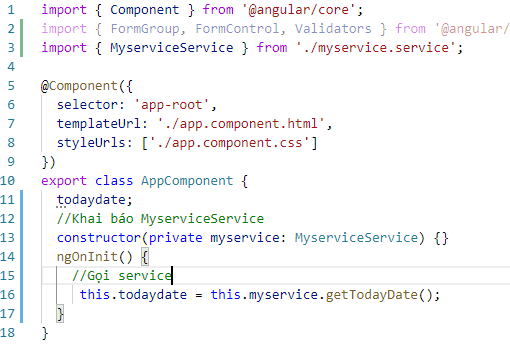
Thay đổi nội dung của myservice: thêm hàm getTodayDate() để lấy ra ngày hôm nay như sau:



1. Sử dụng service:

Bước 1: Thay đổi constructor - thêm service vào component bằng cách khai báo như parameter.

Bước 2: Gọi service trong hàm tạo.



1. HttpService:

Http Service giúp chúng ta thực hiện các phương thức **GET**, **POST** với dữ liệu thông qua giao thức HTTP. Để sử dụng Http Service, chúng ta cần import **http module** .

Để sử dụng **http service**, trước tiên ta cần thực hiện 2 bước:

Bước 1: Import HttpClientModule vào module



Bước 2: Khai báo và sử dụng HttpClient trong component



Phương thức this.http.get() tạo ra 1 get request đến địa chỉ <http://jsonplaceholder.typicode.com/users>. Khi dữ liệu trả về, ta thực hiện xử lý dữ liệu thông qua phương thức subscribe.

1. **ASP.NET CORE và Angular 8 qua Web API:**
2. **ASP.Net Core là gì?**

**ASP.NET Core**là mã nguồn mở mới và framework đa nền tảng (cross-platform) cho việc xây dựng những ứng dụng dựa trên kết nối đám mây, giống như web app cho mobile, … .

1. **Cài đặt môi trường:**

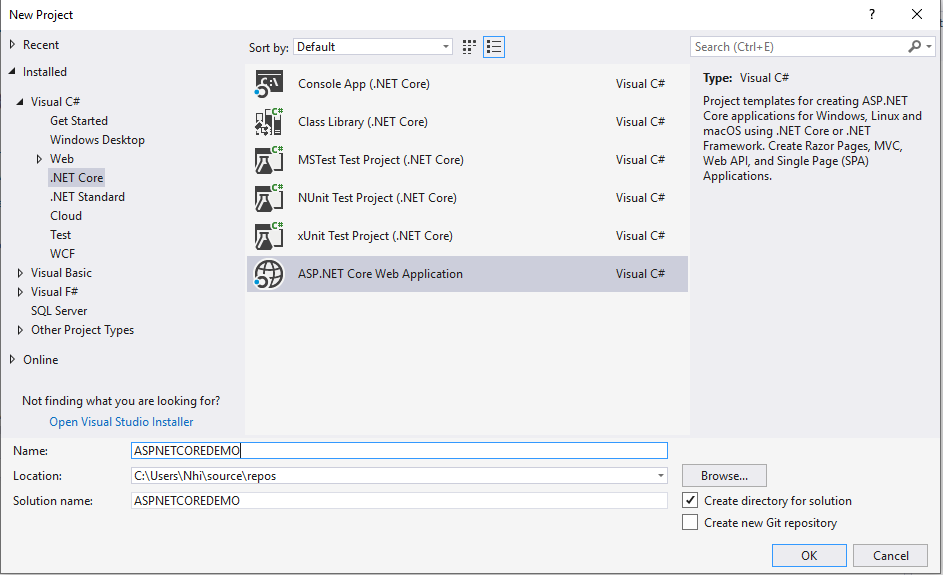
Để làm quen ASP.Net Core cần cài đặt những ứng dụng sau:

* Visual Studio 2017
* .NET Core 1.0.1
* Nodejs

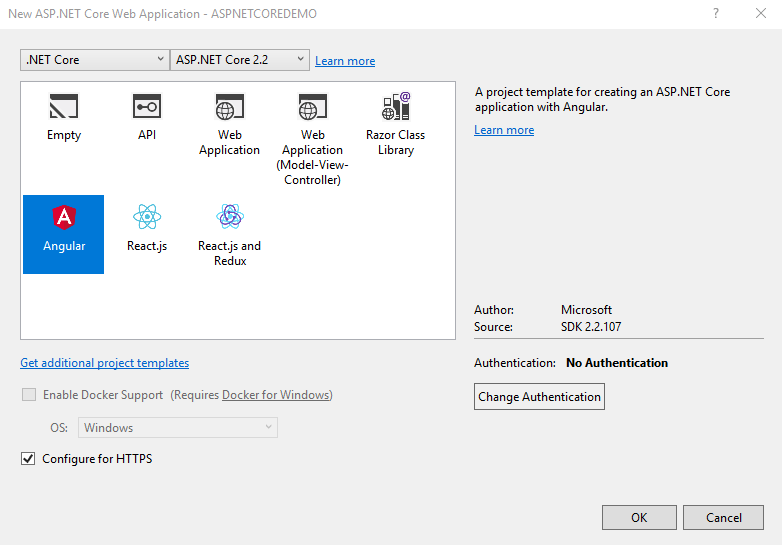
1. **Kết hợp ASP.Net Core và Angular 8 qua web API với Microsoft Visual Studio:**

Sau khi cài đặt xong các điều kiện tiên quyết ở trên, trên màn hình desktop click chọn **Start** =>**Programs** =>**Visual Studio 2017** =>**Visual Studio 2017**.

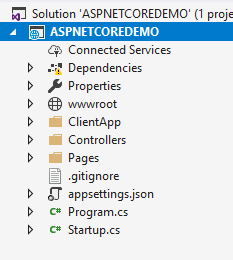
Click chọn **New** =>**Project. Select Web** =>**ASP.NET Core Web Application**. Nhập tên project rồi click chọn **OK**



Cửa sổ mới hiện lên, chọn app Angular, sau đó nhấn OK.

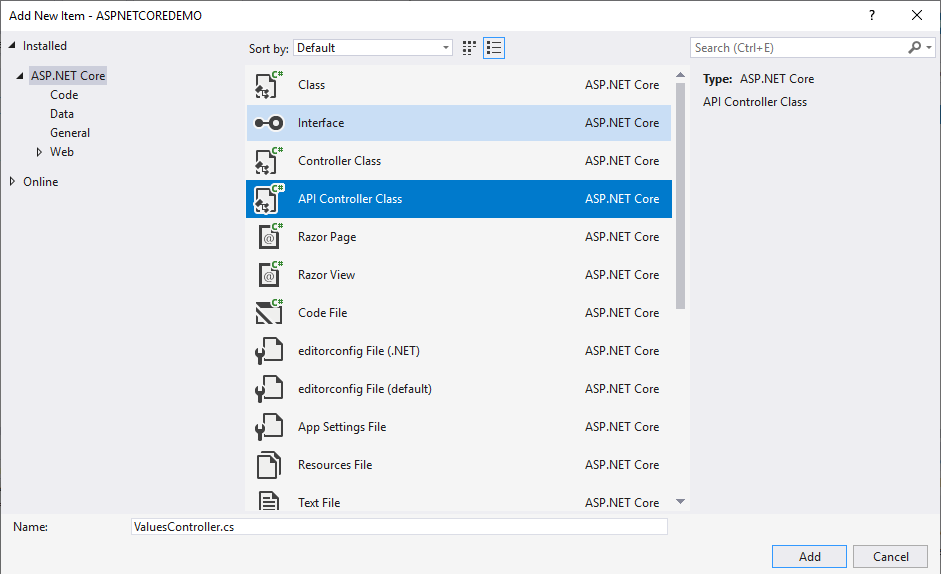


Project sẽ được khởi tạo. Cấu trúc thư mục của project sẽ như sau:



Trong đó, thư mục ClientApp chính là chương trình Angular. Asp.net core kết hợp với Angular để đổ dữ liệu ra phía client.

Tiếp theo, chúng ta tạo sẽ viết web API. Tạo mới một API controller có tên valuesController:



Web API sinh ra 4 phương thức: Get, Post, Put, và Delete.

- Get là yêu cầu dữ liệu. (lựa chọn)

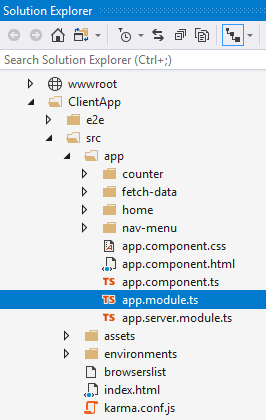
- Post là tạo ra một dữ liệu. (chèn)

- Put là cập nhật dữ liệu.

- Delete là xóa dữ liệu.

Vì phải cần hiển thị kết quả Web API trong ứng dụng Angular nên sẽ có HTTPmodule trong file app.module.

Mở file app.module.





Bây giờ là làm việc với Angular Component để liên kết với Web API và nhận kết quả JSON để liên kết với file HTML.

Mở file Angular Component và chỉnh sửa:



Cụ thể, trong file app component, ta cần import vào HttpClient để lấy dữ liệu từ api của values controller. Trong lớp AppComponent, ta khai báo HttpClient trong constructor và gọi module trong ngOnInit qua hàm this.\_httpService.get

Việc còn lại là hiển thị dữ liệu lên màn hình client, ta chỉnh sửa file appcomponent.html bằng cách data binding như đã đề cập ở phía trên.

Đây là sự kết hợp giữa ASP.Net Core và Angular 8 trong Web API.