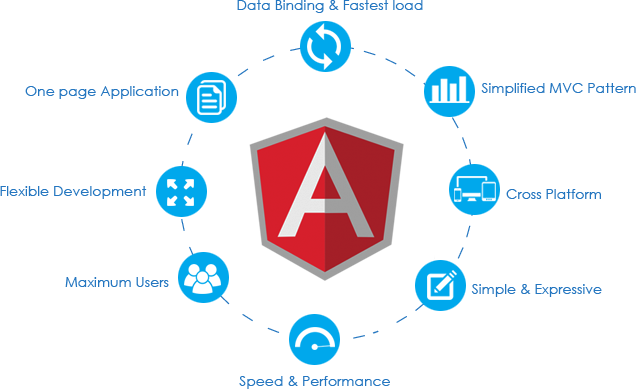
**SEMINAR KHOA HỌC WEB**

**Đề tài: Tìm hiểu framework Angular**

****

**Sinh viên thực hiện:**

**1312037 – Võ Ngọc Bảo**

**1312489 - Nguyễn Ngọc Son**

**Giảng viên hướng dẫn:**

**Thầy Trần Trung Kiên**

**Thầy Phạm Trọng Nghĩa**

**Thầy Lê Ngọc Thành**

Mục lục

[1 What – Angular là gì 3](#_Toc532416928)

[2 The History of Angular 3](#_Toc532416929)

[3 Why – Tại sao nên sử dụng Angular 4](#_Toc532416930)

[4 How 4](#_Toc532416931)

[4.1 Architecture – Kiến trúc Angular 4](#_Toc532416932)

[4.1.1 Overview 4](#_Toc532416933)

[4.1.2 Module 4](#_Toc532416934)

[4.1.3 Angular Library 5](#_Toc532416935)

[4.1.4 Component 5](#_Toc532416936)

[4.1.5 Data binding 6](#_Toc532416937)

[4.1.6 Directive 6](#_Toc532416938)

[4.1.7 Services and dependency injection 6](#_Toc532416939)

[4.1.8 Routing 6](#_Toc532416940)

[4.2 Form 6](#_Toc532416941)

[4.2.1 Reactive Forms 6](#_Toc532416942)

[4.2.2 Template-driven forms 6](#_Toc532416943)

[4.2.3 Form Validation 6](#_Toc532416944)

[4.2.4 Dynamic Forms 6](#_Toc532416945)

[4.3 Observables & The RxJS library 6](#_Toc532416946)

[4.4 Angular Unit Testing 6](#_Toc532416947)

[4.5 Setup and Deployment 6](#_Toc532416948)

[4.5.1 Cài đặt Nodejs 6](#_Toc532416949)

[4.5.2 TypeScript 6](#_Toc532416950)

[4.5.3 Sử dụng Visual Studio Code để tạo project 7](#_Toc532416951)

[4.6 Angular CLI Command 10](#_Toc532416952)

[4.7 Demo 10](#_Toc532416953)

[5 References – Tài liệu tham khảo 10](#_Toc532416954)

# What – Angular là gì

* Angular JS là một **framework** được tạo ra để xây dựng các ứng dụng web động (dynamic web app, **Single Page Application – SPA …)**.
* Angular (từ version 2 trở lên) hỗ trợ ES5, ES6, TypeScript. Hiệu năng nhanh gấp nhiều lần so với AngularJS

# The History of Angular

* 2010: AngularJS được phát triển từ năm 2009 bởi Misko Hevery và Adam Abrons.
* 23/10/2014: sự ra đời của Angular tận dụng sức mạnh của TypeScript và kế thừa các cấu trúc của AngularJS như declarative templates, dependency injection.
* LTS(Long Term Support) của AngularJS sẽ kết thúc vào **June 30, 2021**. Angular Team khuyến khích nên sử dụng Angular(version 2.x trở đi) để xây dựng các dự án hiện nay.
* AngularJS(version 1.x)

|  |  |
| --- | --- |
| **Version** | **Release Date (dd/MM/yyyy)** |
| 1.7.5 | 04/10/2018 |
| 1.7.4 | 07/09/2018 |
| 1.7.3 | 03/08/2018 |
| 1.7.2 | 12/06/2018 |
| 1.7.1 | 08/06/2018 |
| 1.7.0 | 11/05/2018 |
| 1.6.10 | 18/04/2018 |
| 1.6.9 | 02/02/2018 |
| 1.6.8 | 22/12/2017 |
| 1.6.7 | 25/11/2017 |
| 1.6.6 | 18/08/2017 |

* Angular 2.0

|  |  |
| --- | --- |
| **Version** | **Release Date (dd/MM/yyyy)** |
| 9.0.0 | 09/2019 – 10/2019 |
| 8.0.0 | 03/2019 – 04/2019 |
| 7.0.0 | 18/10/2018 |
| 6.0.0 | 05/05/2018 |
| 5.0.0 | 01/11/2017 |
| 4.0.0 | 12/12/2016 |
| 2.0.0 | 23/10/2014 |

# Why – Tại sao nên sử dụng Angular

* Tương thích với nhiều nền tảng: web, mobile web, native mobile and native desktop:
  + kết hợp với các thư viện/framework như NativeScript hay React Native giờ đây bạn có thể tạo ứng dụng mobile một cách dễ dàng với cùng một codebase
* Nâng cao tốc độ và hiệu suất:
  + Typescript giúp code rõ ràng, dễ học, dễ debugs.
  + Angular CLI giúp tạo dự án, quản lý tập tin và thực hiện nhiều tác vụ khác như testing, bundle và deploy dự án một cách nhanh chóng và hiệu quả
* Kiến trúc rõ ràng:
  + Dựa trên mô hình MVVM (model-view-view/model)
  + Data Binding(OneWay, TwoWay), hỗ trợ **properties binding, event binding**
* Routing and Navigation: phân quyền truy cập một cách dễ dàng
* Document and Community:
  + Có một cộng đồng phát triển rất lớn, document rất đầy đủ và chi tiết [angular.io](https://angular.io)
  + Được phát triển bới gã khổng lồ Google, ngày càng hoàn thiện qua nhiều phiên bản.
* MEAN stack: MongoDB, Express.js, AngularJS và Node.js.

# How

## Architecture – Kiến trúc Angular



Nguồn: <https://angular.io/guide/architecture>

### Overview

### Module

* Angular apps là tập hợp các Angular Modules(NgModules)
* Mỗi Angular apps đều có ít nhất 1 module: root module và có thể có thêm feature Module.
* Mỗi Module là một class với decorator @NgModule
  + Angular cung cấp sẵn nhiều decorators gắn các metadata vào các class để dễ dàng biết được ý nghĩa và cách thức hoạt động(VD: @NgModule, @Component…)
* Việc render HTML templates sẽ được thực hiện bởi Angularized-markup, việc cần làm của một developer là viết các component để quản lý template, thêm mới các xử lý logic bên trong các service, đóng gói component và service thành một module.
* NgModule là một decorator function (single metadata object) có các thuộc tính mô tả các module:
  + **declarations** - Khai báo các View classes thuộc về module. Angular có 3 loại view classes: components, directives, pipes.
  + **exports** - Một tập con của declarations khả dụng trong component templates của các module khác.
  + **imports** - Các module khác được exported các classes cần thiết bởi component templates được declarations bên trong module này.
  + **providers** - Tạo ra các global collection of services có thể truy cập ở mọi ngõ ngách của ứng dụng.
  + **bootstrap** - Main application view, hay còn được gọi là root component, chứa tất cả các app views. Duy nhất **root module** có thuộc tính bootstrap, các child-component phải được khai báo trong boostrap của roor module

### Angular Library

* Mỗi Angular library name đều bắt đầu với tiền tố @angular
  + @angular/core
  + @angular/forms
  + @angular/platform-browser
  + @angular/router

### Component

* Angular application là gồm nhiều component.
* Thành phần component
  + View: Khai báo Template HTML, có thể sử dụng các ng Directive(ngIf, ngFor, ngSwitch …) hoặc tag-component để điều chỉnh view.
  + Class: gồm attribute và method code bằng TypeScript
  + Metadata properties:
    - animations - Danh sách các hiệu ứng của component
    - entryComponents - Danh sách các components được tự động chèn vào view của component.
    - encapsulation - Đóng gói cách sử dụng của component.
    - entryComponents - Danh sách các component được sử dụng trong view của component hiện tại.
    - inputs - Danh sách các tên thuộc lớp dữ liệu ràng buộc đầu vào của component.
    - outputs - Danh sách các tên thuộc lớp dữ liệu đầu ra của component.
    - queries - Cấu hình các câu truy vấn được apply vào component.
    - selector - Selector có thể là attribute selectors, tag selectors, class selectors, id selectors hoặc là tổng hợp của những thứ trên.
    - styleUrls - Danh sách các url dẫn đến stylesheet của component.
    - styles - Stylesheet CSS trực tiếp trên component hiện tại.
    - template - HTML code trực tiếp trên component hiện tại.
    - templateUrl - Url dẫn tới file bên ngoài chứa template của view.

### Data binding

* Data Binding:
  + {{<name>}}: interpolation
  + [<propertyName>]: property binding
  + (<eventName>): event binding
  + [( **ngModel** )]: Two-way data binding

### Directive

* Một directive là một class với decorator @Directive
* Component là một directive với template
* Phân loại directive:
  + Structural directives: thay đổi cấu trúc DOM bằng việc thêm bớt các phần tử trong DOM
  + Attribute directives: thay đổi giao diện và tương tác của các đối tượng hoặc thay đổi directive khác

### Services and dependency injection

* Service là một class thực hiện một chức năng cụ thể:
  + VD: logging servie, data service, tax calculator, application configuration
* Một service liên kết với một component sử dụng DI(dependency injection)

### Routing

* Cho phép mở một giao diện khi người dùng thực hiện tác vụ
  + VD: Click vào button để hiển thị chi tiết sản phẩm …
* Ghi lại lịch sử của browser để xử lý khi người dùng click vào button back, forward
* Thêm tag <base href="/"> vào nội dung thẻ <head>
* Router Views
* Routing Module
* Navigation Guards

## Form

### Reactive Forms

### Template-driven forms

### Form Validation

### Dynamic Forms

## Observables & The RxJS library

* Observables: là một tập hợp các callbacks tương ứng cho việc lắng nghe các giá trị (next, error, hay complete) được gửi đến bởi Observable.
* Subscription: là kết quả có được sau khi thực hiện một Observable, nó thường dùng cho việc hủy việc tiếp tục xử lý.
* RxJS là viết tắt của Reactive Extensions Library for JavaScript

## Setup and Deployment

### Cài đặt Nodejs

* Tải và cài đặt Nodejs tại <https://nodejs.org/en/download/>

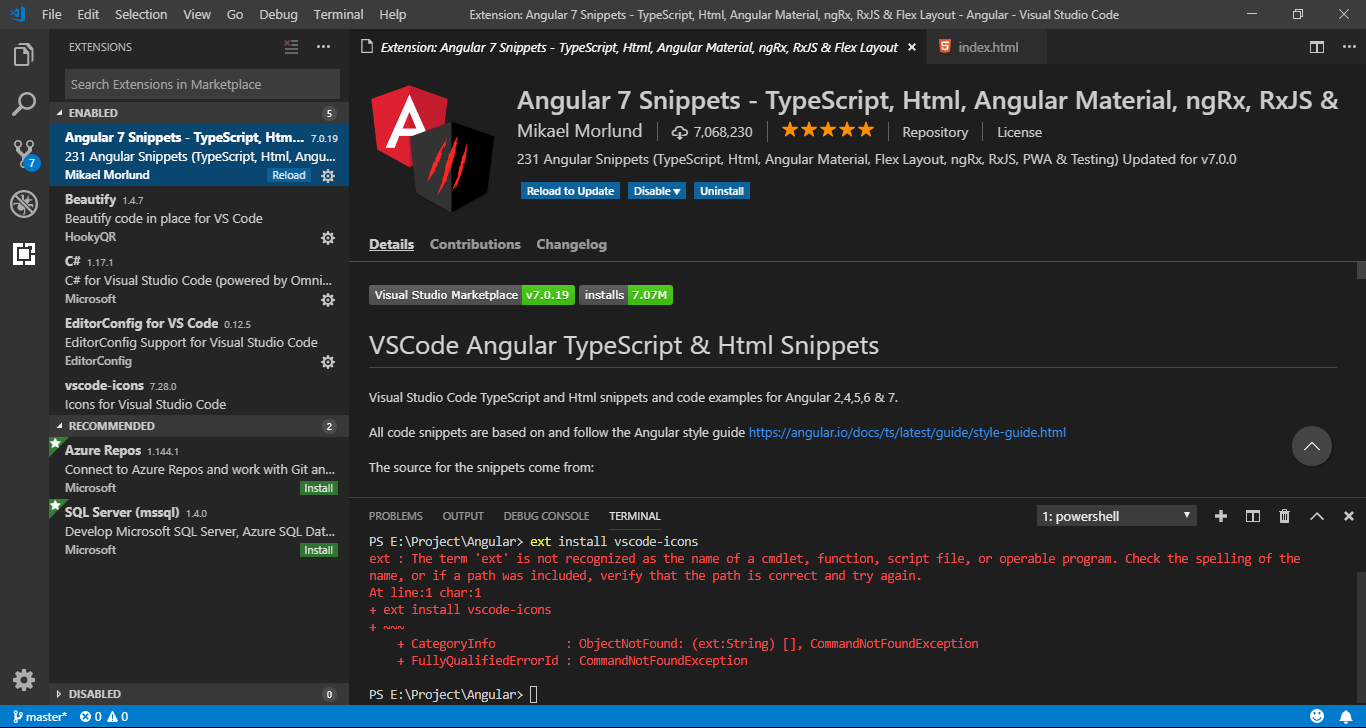
### TypeScript

* Mở command Prompt/Terminal
* Cài đặt TypeScript qua npm bằng lệnh

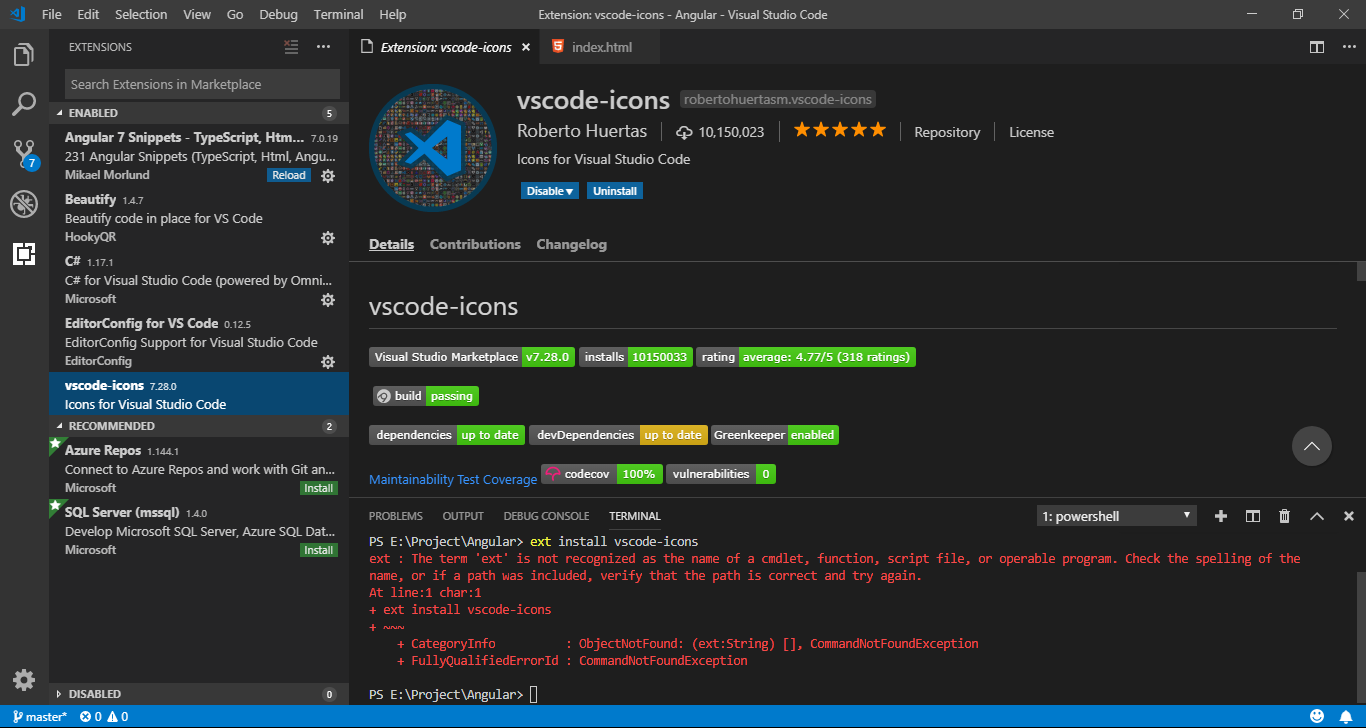
npm install -g typescript

### Sử dụng Visual Studio Code để tạo project

* Thêm extension “Angular 7 Snippets - TypeScript, Html, Angular Material, ngRx, RxJS & Flex Layout”

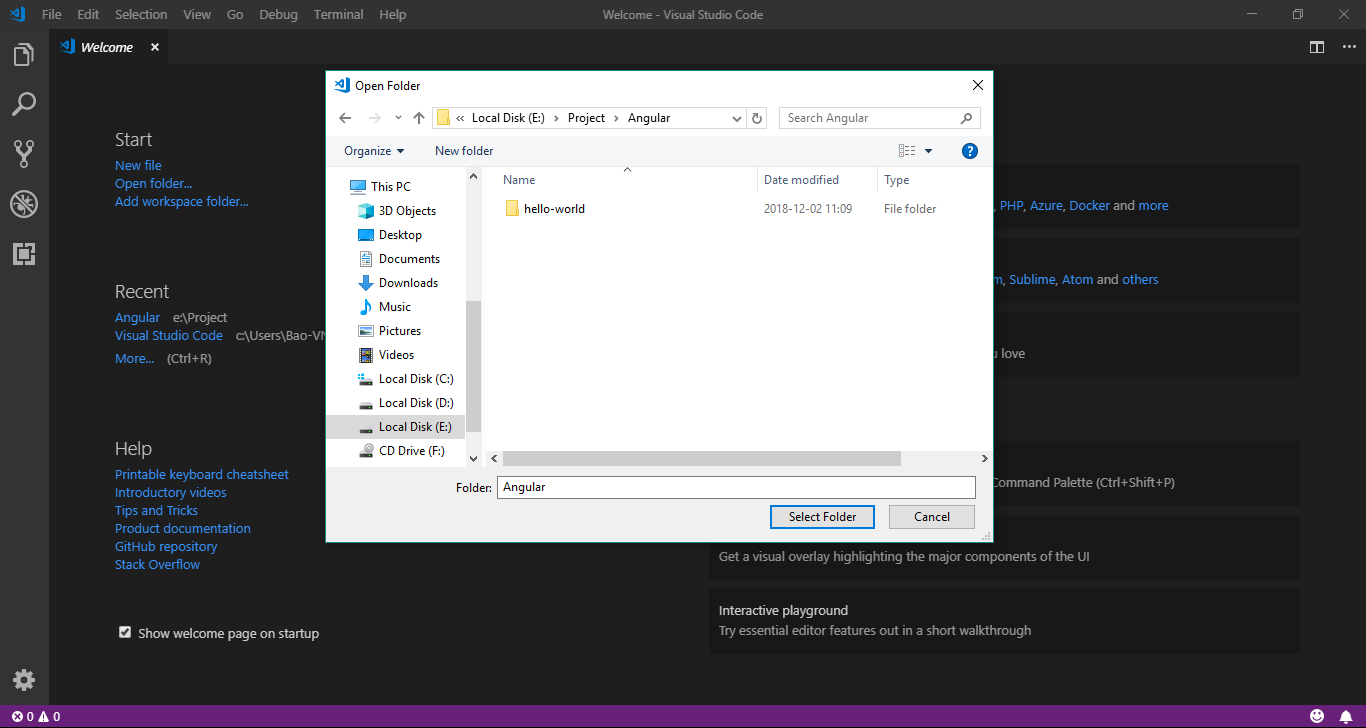


* Thêm extension “vscode-icon”: Để hiển thị icon file, dễ dàng quản lý file

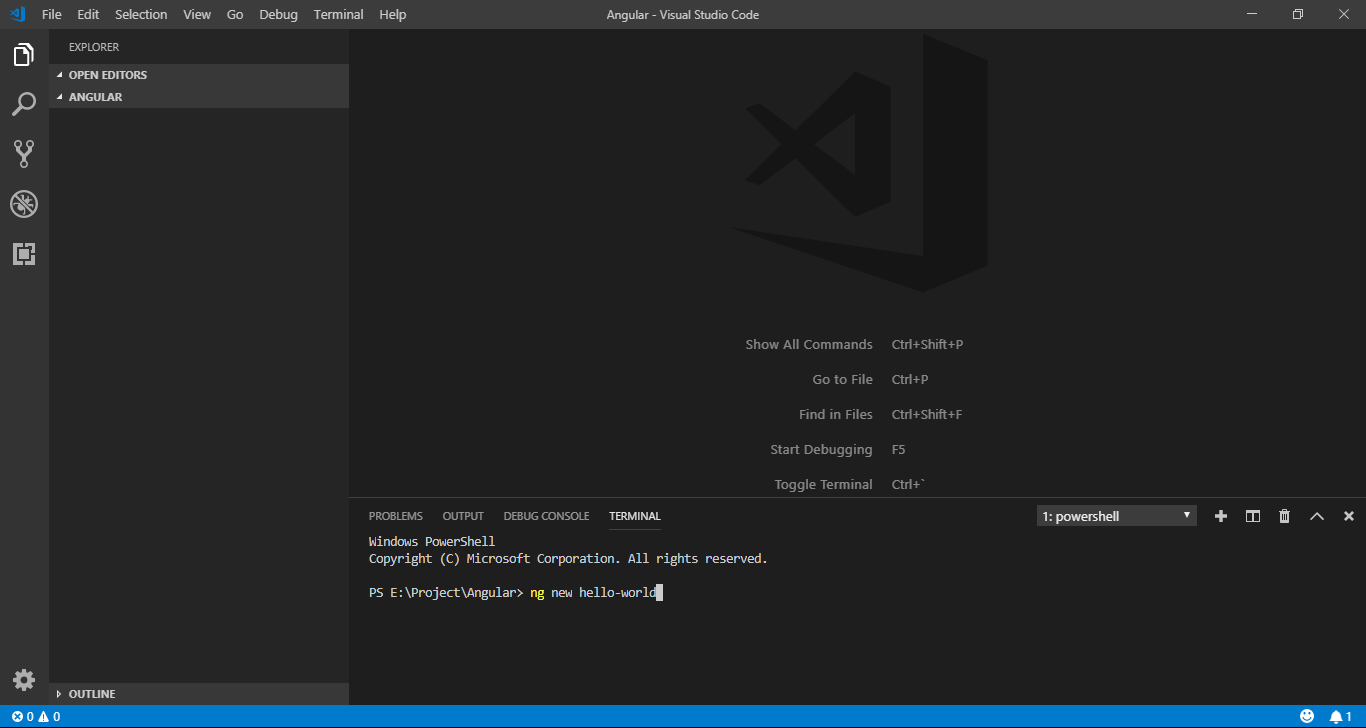


* Tạo project bằng Angular CLI:

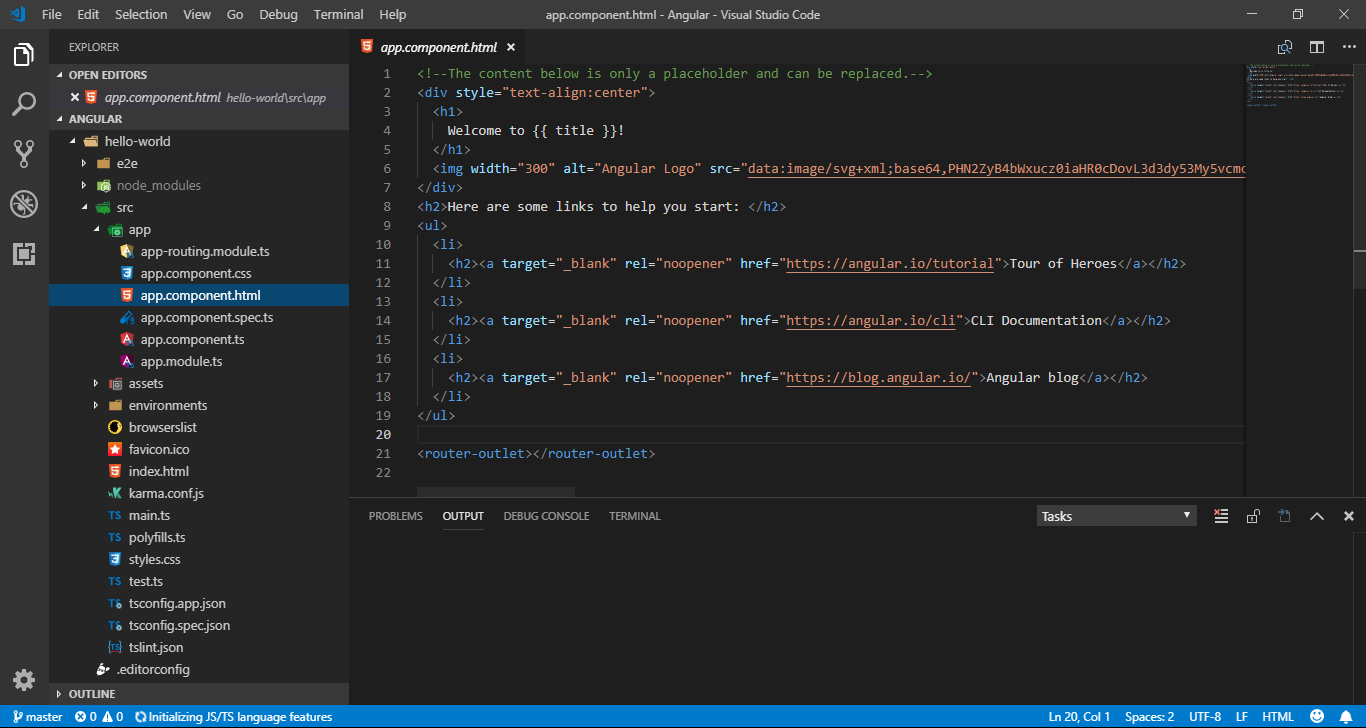
1. Chọn folder muốn tạo project



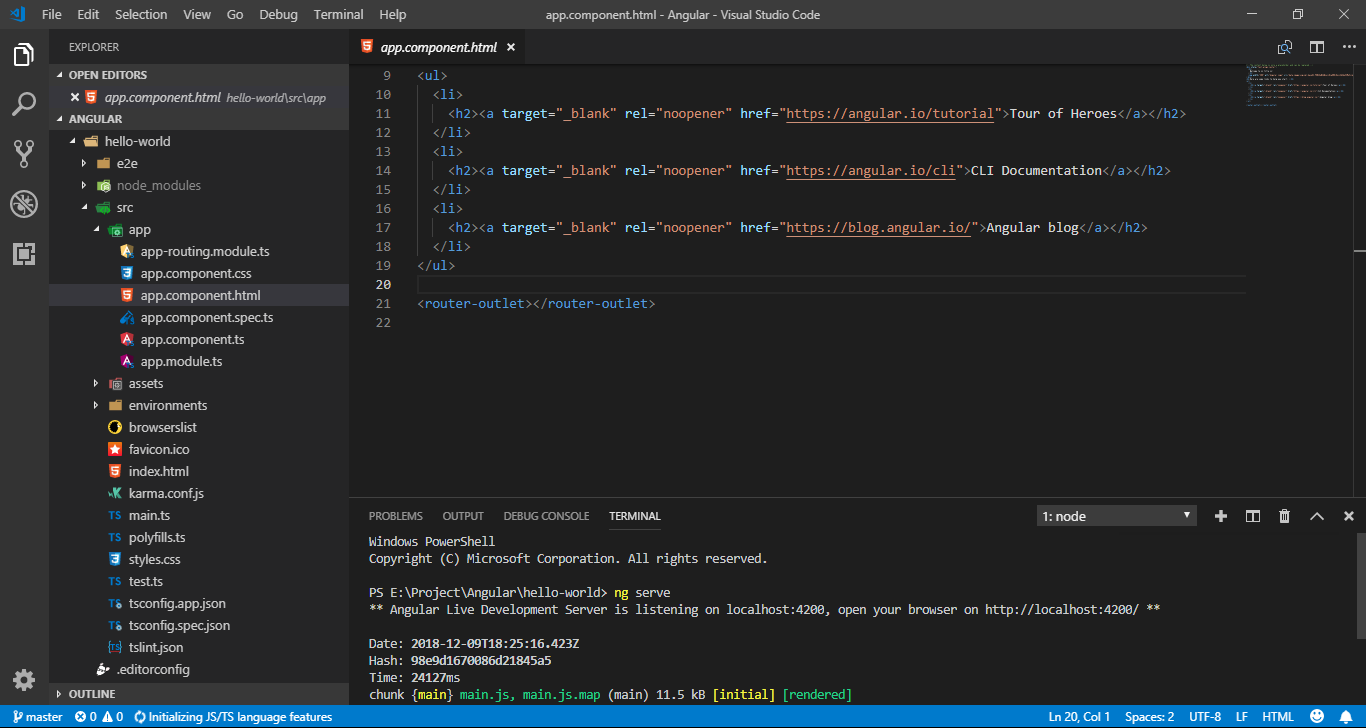
1. Chạy terminal của VS Code và gõ lệnh: ng **new** <*name*>

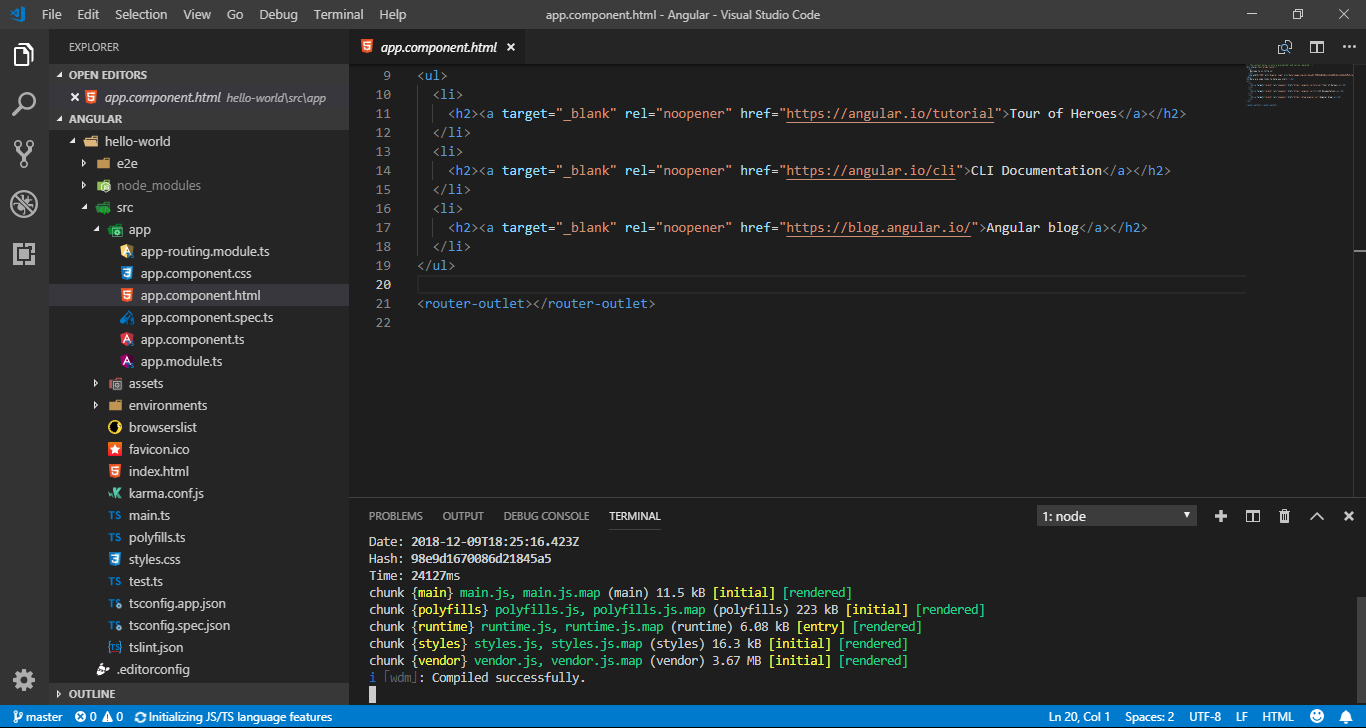


1. Khởi tạo thành công project Angular

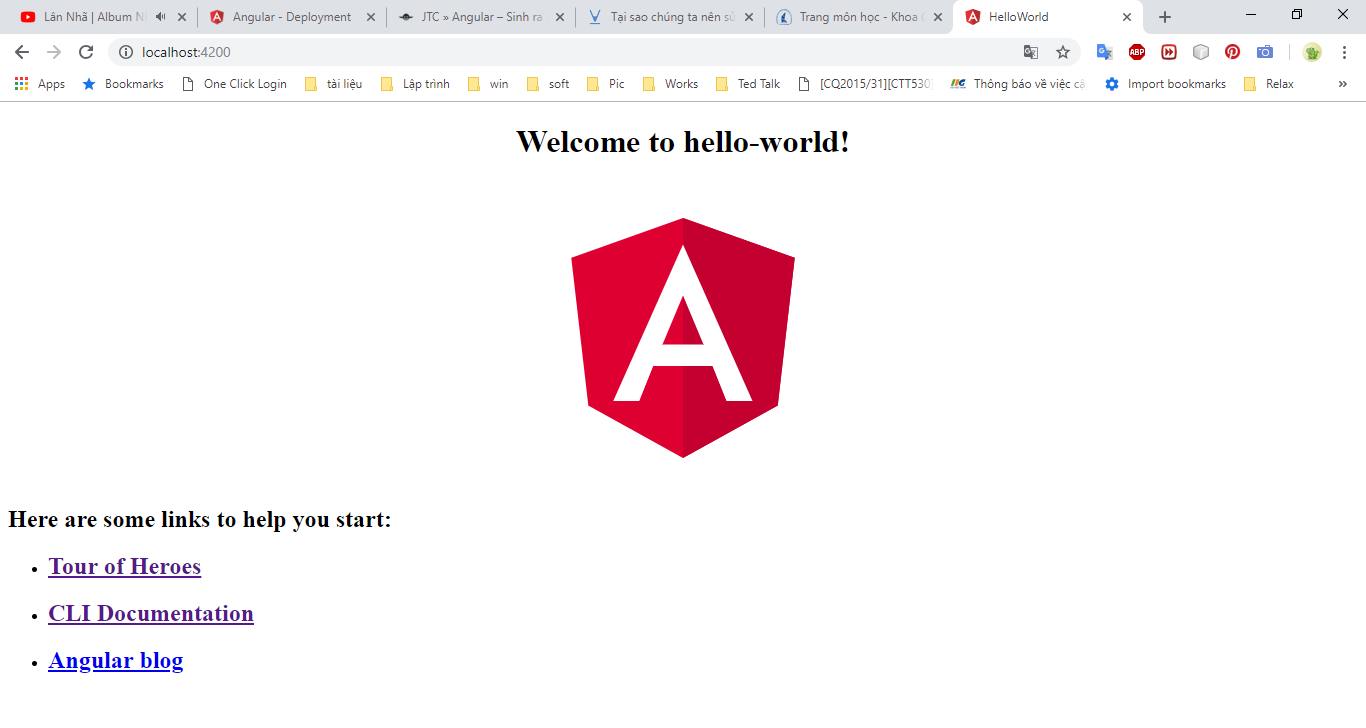


1. Build project ng serve





1. Mở browser, truy cập địa chỉ: <http://localhost:4200/>



## Angular CLI Command

## Demo

# References – Tài liệu tham khảo

* Angular Tutorial: <https://angular.io>
* Series video Angular tutorial of Codevolution: <https://www.youtube.com/watch?v=0eWrpsCLMJQ&index=1&list=PLC3y8-rFHvwhBRAgFinJR8KHIrCdTkZcZ>
* <https://o7planning.org/vi/12077/gioi-thieu-ve-angularjs-va-angular>
* Kipalog: <https://kipalog.com/posts/Gioi-thieu-series-hoc-lap-trinh-Angular--Angular-2>