# Tại sao nên sử dụng Angular2?

Hiện nay có rất nhiều các framework JavaScript được chọn, mỗi thứ đều có điểm mạnh yếu riêng. Nhiều người cảm thấy tốt khi sử dụng Angular 1.x nhưng Angular2 đã được cải tiến và chạy nhanh hơn, dễ dàng mở rộng hơn và hiện đại hơn. Các nhà phát triển đã từng thấy giá trị của Angular 1.x, sẽ tìm thấy nhiều lợi ích hơn nữa từ Angular2.

## Lợi ích của Angular2

Phiên bản đầu tiên của Angular đã được đưa ra cho các nhà phát triển một công cụ để xây dựng hệ thống ứng dụng lớn với JavaScript nhưng nó đã bộc lộ một số hạn chế. Angular2 đã được ra đời từ những phản hồi của cộng đồng.

## Angular2 sẽ dễ dàng hơn

Angular2 được phát triển trên nền tảng hiện đại hơn, nhiều tính năng, và dễ dàng cho người mới học hơn là Angular 1.x và cũng dễ dàng hơn cho các chiến binh lâu năm.

Với Angular 1, các nhà phát triển  đã biết về  sự khác nhau giữa Controllers, Services, Factories, Providers và các khái niệm khác có thể gây ra sự khó hiểu đặc biệt là các coder mới.

Angular 2 là một framework với sự sắp xếp hợp lý hơn cho phép lập trình viên tập trung vào việc tạo lên các class JavaScript đơn giản. Views và các controllers sẽ được thay thế bởi Components, thứ mà có thể mô tả là một sự tinh chế mới của directives. Ngay cả các lập trình viên có kinh nghiệm với Angular cũng không chắc là có thể biết tất cả những khả năng của directives. Angular 2 component đã tăng khả năng đọc đáng kể, các API của chúng giảm đáng kể các biệt ngữ hơn là Angular 1.x directive.

## TypeScript

Angular2 được viết với TypeScript, một superset của JavaScript, nó triển khai nhiều tính năng của ES2016 và các tính năng của TypeScript.

Bằng cách tập trung tạo nên một framework dễ dàng hơn trong việc xử lý, Angular2 cho phép hệ sinh thái phát triển phong phú hơn. Các lập trình viên có thể sử dụng các trình soạn thảo tinh vi hơn với các tính năng gợi ý code hay tự động hoàn thành code. Những tính năng này giúp giảm thời gian học Angular2. May mắn hơn nữa cho các nhà phát triển ES5 truyền thống rằng không nhất thiết phải viết ES6 hay TypeScript, các coder có thể vẫn viết JavaScript thuần mà không cần biên dịch.

## Quen thuộc

Mặc dù được viết lại hoàn toàn, Angular2 vẫn giữ lại các khái niệm cốt lõi và các quy tắc của Angular1.x. Điều đó có nghĩa là các lập trình viên giỏi vói Angular1 sẽ dễ dàng  chuyển đổi sang Angular2 hơn là các thư viện khác như ReactJS....

## Hiệu năng và tương thích mobile

Angular2 được thiết kế cho mobile từ ban đầu. Bỏ qua một bên các vấn đề về sức mạnh xử lý, các thiết bị di động có các tính năng khác và sự giới hạn đó phân biệt chúng với máy tính truyền thống. Giao diện chạm, màn hình nhỏ, phần cứng đã được tính toán trong Angular2. Máy tính để bàn cũng sẽ được cải tiến về hiệu suất và khả năng đáp ứng.

Angular2 giống React và các framework hiện đại khác, có thể tăng hiệu năng bằng cách render ra HTML trên server hoặc ngay cả trên browser. Phụ thuộc vào thiết kế cấu trúc giúp cho trải nghiệm người dùng tốt hơn.

Yêu cầu về hiệu năng không bao giờ là dừng lại với việc tiền xử lý. Angular 2 có khả năng tích hợp với mobile bằng cách tích hợp với NativeScript, một thư viện mã nguồn mở là cậu nỗi giữa JavaScript và mobile. Bổ sung thêm, Ionic cũng đang được phát triển trên Angular2 cung cấp một cách khác giúp tăng hiệu quả của ứng dụng trên mobile.

## Kiến trúc dự án và bảo trì

Phiên bản đầu tiên của Angular cung cấp cho các lập trình viên một thư viện mềm dẻo để phát triển ứng dụng. Đây là sự thay đổi đáng kể cho các lập trình viên web, và trong khi framework rất hữu ích thì nó là một bằng chứng cho sự linh hoạt đó. Qua thời gian, các best practice từ cộng đồng được tán thành rộng rãi.

Angular 1.x đã cố gắng làm việc xung quanh các hạn chế trên trình duyệt liên quan đến JavaScript. Điều này được giải quyết bằng cách cung cấp một hệ thống các module sử dụng cơ chế Dependency Injection. Điều này thật tuyệt tuy nhiên có một vài vấn đề.

Angular 2.x sử dụng hệ thống module ECMA Script 6 (ES6), và các công cụ đóng gói hiện đại như Webpack hoặc SystemJS. Module được tùy biến theo cách riêng của Angular để dễ dàng viết các Generic cho JS để cắm vào Angular. Hệ thống module mới giúp dễ dàng phát triển các dự án lớn hiệu quả hơn.

## Tính năng mới

Một vài tính năng thú vị trên Angular2

* Form Builder
* Change Detection
* Templating
* Routing
* Annotations
* Observables
* Shadow DOM

## Sự khác nhau giữa AngularJS và Angular2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Angular 1.x** | Angular 1.x  Best  Practices | **Kiến trúc chuyển tiếp** | Angular 2 |
| Scope lồng nhau  ("$scope") | Sử dụng nặng | Tránh | Tránh | Đã bỏ |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Angular 2 căn bản**

# Bài 01: Giới thiệu về Angular 2

Angular 2 là 1 framework phát triển trên nền JavaScript của Google, kế thừa các đặc điểm của AngularJS và phát triển một phương thức tiếp cận việc xây dựng ứng dụng hoàn toàn mới, phương pháp hướng Component.  
Video này sẽ giúp các bạn có cái nhìn tổng quan về các tính năng và đặc tính của Angular2.

**Yêu cầu trước tiên:**

Bạn cần phải có một sự hiểu biết cơ bản về JavaScript và editor. Nếu phát triển các ứng dụng dựa trên web sử dụng Angular 2, sẽ tốt hơn nếu bạn có một sự hiểu biết về công nghệ web khác như HTML, CSS, AJAX, AngularJS(bạn có thể không cần biết AngularJS vẫn có thể học được Angular2, tuy nhiên Angular2 được kế thừa rất nhiều ưu điểm từ AngularJS nên nếu bạn biết AngularJS sẽ là một lời thế để có thể dễ dàng tiếp cận Angular2 hơn)…

**Angular 2 là gì?**

* Angular 2 là phiên bản tiếp theo của framework phát triển dựa trên JavaScript của Google.
* Angular 2 được thiết kế để xây dựng các ứng dụng phức tạp cho trình duyệt.
* Ngược lại với phiên bản 1.x, Angular 2 giới thiệu một khái niệm hoàn toàn mới của việc xây dựng các ứng dụng web.
* Phiên bản beta của Angular 2 đã được phát hành vào tháng Ba năm 2014.

**Tại sao phải sử dụng Angular 2?**

Angular 2 là đơn giản hơn so với Angular 1 và khái niệm của nó làm cho nó dễ dàng hơn để hiểu. Bạn có thể cập nhật các bộ dữ liệu lớn với bộ nhớ tối thiểu. Nó sẽ tăng tốc độ tải ban đầu thông qua cơ chế rendering trên server

**Đặc tính**

* Angular 2 là nhanh hơn và dễ dàng hơn so với Angular 1.
* Nó hỗ trợ tất các phiên bản mới của trình duyệt và cũng hỗ trợ các trình duyệt cũ bao gồm IE9 + và Android 4.1 trở lên.
* Nó là một framework đa nền tảng.
* Angular 2 chủ yếu tập trung vào các ứng dụng di động.
* Cấu trúc mã là rất đơn giản hơn so với phiên bản trước.
* Typing sử dụng TypeScript, Dart, JavaScript

**Điểm mạnh**

* Nếu một ứng dụng được tải nặng, sau đó Angular 2 giữ nó hoàn toàn giao diện người dùng.
* Nó sử dụng máy chủ rendering cho view nhanh trên điện thoại di động.
* Nó hoạt động tốt với ECMAScript và các ngôn ngữ khác để biên dịch JavaScript.
* Nó sử dụng Dependency Injection chạy các ứng dụng mà không cần viết mã quá dài.
* Tất cả mọi thứ sẽ là cách tiếp cận dựa trên Component

**Hạn chế**

Từ Angular 2 là một framework mới được giới thiệu, có ít hỗ trợ bởi cộng đồng. Phải mất thời gian để tìm hiểu nếu bạn là người mới học Angular 2.

# Bài 02: Kiến trúc Angular 2

Bài này chúng ta sẽ cùng tìm hiểu về kiến trúc của Angular 2 bao gồm các thành phần chính tạo nên một ứng dụng Angular 2 và cơ chế Dependency Injection được tích hợp sẵn trong Angular 2.0

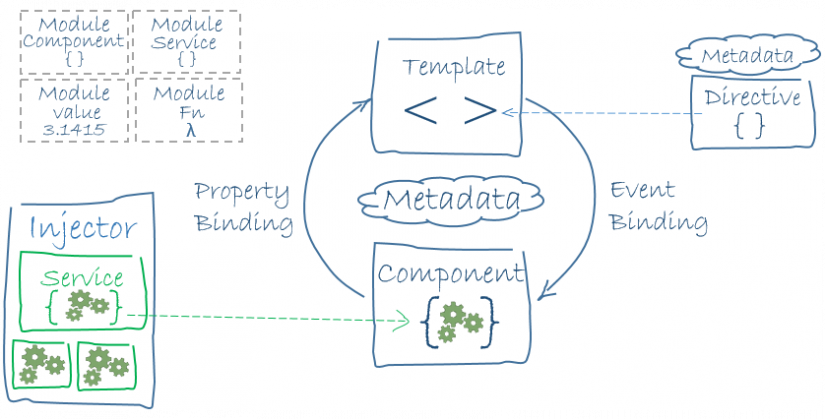
Tham khảo:

<http://phocode.com/javascript/lap-trinh-web-voi-angularjs-2/angular-kien-truc/>

<https://github.com/truonganhhoang/int3507-2016/wiki/%5BXYZ%5D-Angular-2>

<http://www.c-sharp.vn/angular/angular-2-series-architecture-ef3539>

Kiến trúc tổng quan của 1 ứng dụng angular2 được thể hiện trong hình ảnh sau:



Hình ảnh này mình đã lấy tại trang chủ của angular.io của Angular

Quan sát sơ đồ trên chúng ta có 1 thành phần chính là component. Tất cả các ứng dụng angular2 có thành phần chính là component,　bởi nó đều base trên hướng tiếp cận component.

Và trong component có 3 thành phần

+ Thành phần đầu tiên: class component

Component có chức năng điều khiển việc hiển thị, tức là điều khiển View, vậy bạn có thể hình dung Component chính là một Controller trong mô hình MVC…

Ví dụ một đoạn code component:

+ Thành phần thứ 2: metadata, nó dùng các decorater để add thêm thông tin cho class component

+ Thành phần thứ 3: template, hay cụ thể hơn nó là file html dùng để hiển thị view cho class component

Metadata là phần kết nối giữa template và component. nó sẽ quy định file template nào sẽ hiển thị cho component này. Để dễ tưởng tượng thì trong các framework php ta sẽ dễ dàng lien tưởng đến component như là các controller, template là các view hiển thị và metadata chính là thành phần routing điều hướng hiển thị của controller với template.

Component và template được lien kết với nhau qua cơ chế Event Binding và Property Binding. Ngoài ra ta cũng có các directive là các thành phần mở rộng của template html để thực hiện một thao tác nào đó, một tác vụ nào đó.

Một ứng dụng Angular được xây dựng từ 8 thành phần sau đây: Module, Component, Template, Metadata, Data Binding, Directive, Service, Dependency Injection.

## Module

Mỗi ứng dụng Angular được gọi là một module và bản thân Angular có riêng một module dùng để quản lý các module khác có tên là Root Module hay NgModule. Root Module thường được đặt tên là AppModule, ngoài root ra thì tùy ứng dụng mà sẽ có thêm các module khác, chúng ta sẽ tìm hiểu về root module trong bài sau.

Chúng ta khai báo một module bằng cách dùng từ khóa @NgModule. Các từ khóa như @NgModule này là các hàm dùng để chỉnh sửa các lớp của Javascript. Bên trong từ khóa @NgModule chúng ta khai báo các tham số sau đây:

* declarations: tên lớp view thuộc về module này
* exports: danh sách tên các module hoặc component có thể sử dụng module này
* imports: tên các module sẽ được dùng từ module này
* providers: tên các service sẽ được dùng từ module này, chúng ta sẽ tìm hiểu về service sau
* bootstrap: tên lớp view dành cho root module, chỉ có root module mới thiết lập tham số này

# Bài 03: Cài đặt môi trường và tạo Hello World app

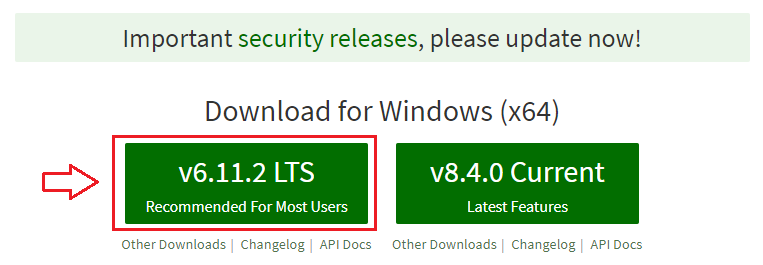
Trong bài số 3, chúng ta sẽ học về cách cài đặt môi trường và cài đặt template. Nội dung của bài học, chúng ta có các phần như là:

* + Cài đặt Nodejs và Visual Code
  + Cài đặt Angular2 bằng Basic Template

1. Cài đặt NodeJS và Visual Code

Đầu tiên chúng ta cài đặt Nodejs tại trang chủ của nodejs: <https://nodejs.org/en/>

Chúng ta nên sử dụng phiên bản “recommender for most users”



Nodejs là một engine, được hỗ trợ cài đặt cả npm chủ yếu dung để get,set các module của angular về để sử dụng. Bản thân angular2 cũng là 1 module của npm.

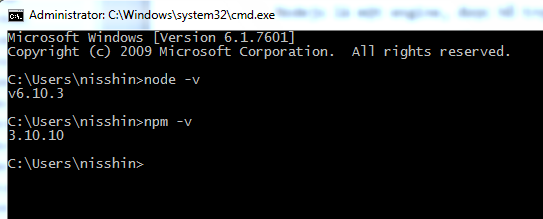
Tiếp theo ta download Visual Code tại: <https://code.visualstudio.com/>

2. Kiểm tra version của nodejs và npm

Sau khi cài đặt nodejs thì ta có thể kiểm tra version của nó bằng cách mở cmd trên Windows và thực hiện lệnh command như sau:

#> node –v

#> npm –v



# Bài 04: Hiểu về các thành phần của ứng dụng Angular 2

# Bài 05: Tạo mới 1 component

# Bài 06: Áp dụng CSS cho component

# Bài 07: One way binding

# Bài 08: Binding class và style cho element

# Bài 09: Event binding

# Bài 10: Two way binding

# Bài 11: Stuctural directives

# Bài 12: Binding class và style cho element

**Bài 12**: Binding class và style cho element

**Bài 12**: Binding class và style cho element

**Bài 12**: Binding class và style cho element

**Bài 12**: Binding class và style cho element