

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

Khoa Công Nghệ Phần Mềm



ĐỒ ÁN CHUYÊN NGÀNH

**LỚP: SE112.I11**

**ĐỀ TÀI: TÌM HIỂU ANGULAR 2, 4**

**VÀ XÂY DỰNG ỨNG DỤNG MINH HỌA**

Giảng viên hướng dẫn:

ThS. Hầu Nguyễn Thành Nam

Sinh viên thực hiện:

Trần Gia Sang - 13520700

*Tp. Hồ Chí Minh, ngày 19 tháng 11 năm 2017*

Mục lục

[Chương 1: GIỚI THIỆU NODE JS 1](#_Toc498903758)

[NodeJS là gì? 1](#_Toc498903759)

[Đặc điểm của Node.js 1](#_Toc498903760)

[Với Node.js, bạn phải làm mọi thứ 2](#_Toc498903761)

[Ai sử dụng Node.js 2](#_Toc498903762)

[Các thành phần quan trọng trong Node.js 2](#_Toc498903763)

[NodeJs được sử dụng ở đâu 2](#_Toc498903764)

[Nodejs không nên sử dụng ở đâu: 2](#_Toc498903765)

[Tại sao lại sử dụng Node.js? 3](#_Toc498903766)

[Nhược điểm 3](#_Toc498903767)

[Chương 2: GIỚI THIỆU ANGULAR 4](#_Toc498903768)

[Tổng quát 4](#_Toc498903769)

[Yêu cầu trước tiên: 4](#_Toc498903770)

[Angular 2 là gì? 4](#_Toc498903771)

[Tại sao phải sử dụng Angular 2? 4](#_Toc498903772)

[Đặc tính 5](#_Toc498903773)

[Điểm mạnh 5](#_Toc498903774)

[Hạn chế 5](#_Toc498903775)

[Chương 3: ANGULAR 4 CÓ GÌ MỚI? 5](#_Toc498903776)

[1. Tổng quan 5](#_Toc498903777)

[2. New features 5](#_Toc498903778)

[3. Breaking changes 10](#_Toc498903779)

[Deprecated 13](#_Toc498903780)

[Chương 4: XÂY DỰNG ỨNG DỤNG MINH HỌA 13](#_Toc498903781)

[Chương 5: TỔNG KẾT 13](#_Toc498903782)

# GIỚI THIỆU NODE JS

## NodeJS là gì?

NodeJS là một nền tảng Server side được xây dựng dựa trên Javascript Engine (V8 Engine). Node.js được phát triển bởi Ryan Dahl năm 2009 và phiên bản cuối cùng là v0.10.36. Định nghĩa NodeJs bởi tài liệu chính thức như sau:

Node.js là một nền tảng dựa vào Chrome Javascript runtime để xây dựng các ứng dụng nhanh, có độ lớn. Node.js sử dụng các phần phát sinh các sự kiện (event-driven), mô hình non-blocking I/O để tạo ra các ứng dụng nhẹ và hiệu quả cho các ứng dụng về dữ liệu thời gian thực chạy trên các thiết bị phân tán.

NodeJs là một mã nguồn mở, đa nền tảng cho phát triển các ứng dụng phía Server và các ứng dụng liên quan đến mạng. Ứng dụng Node.js được viết bằng Javascript và có thể chạy trong môi trường Node.js trên hệ điều hành Window, Linux...

Node.js cũng cung cấp cho chúng ta các module Javascript đa dạng, có thể đơn giản hóa sự phát triển của các ứng dụng web sử dụng Node.js với các phần mở rộng.

Node.js = Môi trường Runtime + Các thư viện Javascript

## Đặc điểm của Node.js

Dưới đây là vài đặc điểm quan trọng biến Node.js trở thành sự lựa chọn hàng đầu trong phát triển phần mềm:

* **Không đồng bộ và Phát sinh sự kiện (Event Driven)**: Tất các các APIs của thư viện Node.js đều không đồng bộ, nghĩa là không blocking (khóa). Nó rất cần thiết vì Node.js không bao giờ đợi một API trả về dự liệu. Server chuyển sang một API sau khi gọi nó và có cơ chế thông báo về Sự kiện của Node.js giúp Server nhận đựa phản hồi từ các API gọi trước đó.
* **Chạy rất nhanh**: Dựa trên V8 Javascript Engine của Google Chrome, thư viện Node.js rất nhanh trong các quá trình thực hiện code.
* **Các tiến trình đơn giản nhưng hiệu năng cao**: Node.js sử dụng một mô hình luồng đơn (single thread) với các sự kiện lặp. Các cơ chế sự kiện giúp Server trả lại các phản hồi với một cách không khóa và tạo cho Server hiệu quả cao ngược lại với các cách truyền thống tạo ra một số lượng luồng hữu hạn để quản lý request. Nodejs sử dụng các chương trình đơn luồng và các chương trình này cung cấp các dịch vụ cho số lượng request nhiều hơn so với các Server truyền thống như Apache HTTP Server.
* **Không đệm**: Ứng dụng Node.js không lưu trữ các dữ liệu buffer.
* **Có giấy phép**: Node.js được phát hành dựa vào MIT License.

## Với Node.js, bạn phải làm mọi thứ

Node.js chỉ là một môi trường – điều này có nghĩa bạn tự phải làm mọi thứ. Sẽ chẳng có bất kỳ máy chủ mặc định nào cả !!! Một đoạn script xử lý tất cả các kết nối với Client. Điều này làm giảm đáng kể số lượng tài nguyên được sử dụng trong ứng dụng.

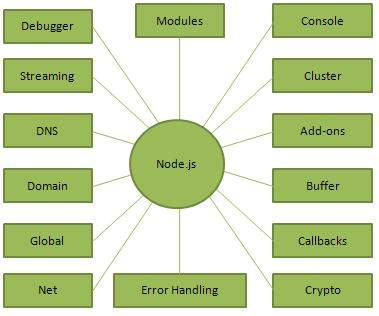
## Ai sử dụng Node.js

Dưới đây là link trên github wiki chứa danh sách các dự án, ứng dụng và các công ty sử dụng Node.js. Trong danh sách này bao gồm eBay, GE, GoDaddy, Microsoft, Paypal, Uber....

* [Project, Application, và Company sử dụng Node.js](https://github.com/joyent/node/wiki/projects,-applications,-and-companies-using-node)

## Các thành phần quan trọng trong Node.js

Lược đồ dưới đây mô tả các thành phần quan trọng của Node.js mà chúng ta sẽ thảo luận ở các chương tiếp theo.



## NodeJs được sử dụng ở đâu

Dưới đây là các lĩnh vực mà Node.js được sử dụng như là một sự lựa chọn hoàn hảo:

* Các ứng dụng về I/O
* Các ứng dựng về luồng dữ liệu
* Các ứng dụng về dữ liệu hướng đến thời gian thực
* Các ứng dụng dựa vào JSON APIs
* Các ứng dụng Single Page Application

## Nodejs không nên sử dụng ở đâu:

Nó không nên sử dụng trong các ứng dụng đòi hỏi về CPU.

## Tại sao lại sử dụng Node.js?

Đầu tiên là ưu điểm về tốc độ thực thi và khả năng mở rộng. Node.js có tốc độ rất nhanh. Đó là một yêu cầu khá quan trọng khi bạn là một startup đang cố gắng tạo ra một sản phẩm lớn và muốn đảm bảo có thể mở rộng nhanh chóng, đáp ứng được một lượng lớn người dùng khi trang web của bạn phát triển lên.

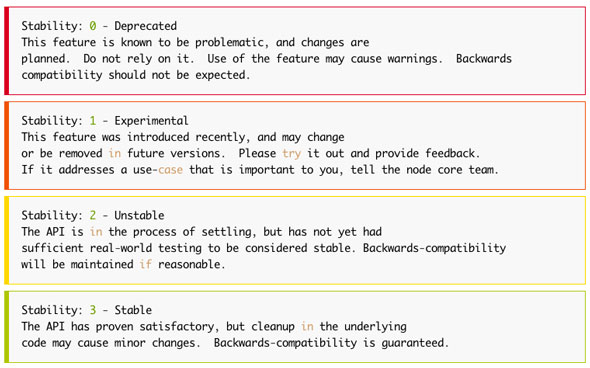
Node.js có thể xử lý hàng ngàn kết nối đồng thời trong khi PHP sẽ chỉ có nước sụp đổ. Bên cạnh các lợi ích về tốc độ thực thi và khả năng mở rộng, có thể bạn cũng đã biết một chút về JavaScript, vì vậy tại sao lại phải phiền toái để học thêm về một ngôn ngữ lập trình hoàn toàn mới như PHP? Và sau đó bạn sẽ có một sự phấn khích khi học về một cái gì đó mới mẻ và gần như chưa được khám phá. Bạn còn nhớ cái cảm giác khi mà một cái gì đó mới xuất hiện và sau đó trở thành phổ biến khắp mọi nơi mà bạn hối tiếc đã không học về nó sớm hơn, và mãi mãi chỉ là người đến sau? Đừng phạm phải sai lầm như vậy lần này nữa. Node.js đang ngày càng trở nên lớn mạnh hơn.

## Nhược điểm

Giống như hầu hết các công nghệ mới, việc triển khai Node.js trên host không phải là điều dễ dàng. Nếu bạn có một web hosting xài chung, bạn không thể đơn giản tải lên một ứng dụng Node.js và mong chờ nó hoạt động tốt. VPS và dedicated server là một sự lựa chọn tốt hơn - bạn có thể cài đặt Node.js trên chúng. Thậm chí dễ hơn là sử dụng một dịch vụ có khả năng mở rộng như là Heroku, và bạn có thể hoàn toàn an tâm để phát triển trang web của mình trên đó - bạn chỉ cần trả tiền khi cần thêm nhiều tài nguyên hơn.

Mặt khác, chúng ta rất dễ cài đặt Node.js chạy cục bộ trên máy tính của bạn sử dụng các hệ điều hành như Windows, Mac hoặc Linux và bắt đầu phát triển ứng dụng ngay lập tức - chỉ việc tải phiên bản Node.js tương ứng tại đây. Một điều quan trọng nên chú ý là Node.js không chỉ đơn giản là một sự thay thế cho Apache - các ứng dụng web đang tồn tại sẽ không có khả năng tương thích, và bạn sẽ làm việc hiệu quả với những ứng dụng phát triển từ đầu.

Một nhược điểm lớn khác của Node.js đó là nó vẫn đang trong giai đoạn phát triển ban đầu, điều này có nghĩa là một số đặc trưng sẽ thay đổi trong quá trình phát triển tiếp theo. Trong thực tế, nếu bạn đọc các tài liệu đi kèm, thì nó bao gồm một chỉ số ổn định (stability index), chỉ số này cho thấy mức độ rủi ro khi bạn sử dụng các đặc trưng hiện có.



# GIỚI THIỆU ANGULAR

## Tổng quát

Angular 2 & 4 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 Angular 2 & 4.

## 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…

## 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 22 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 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.

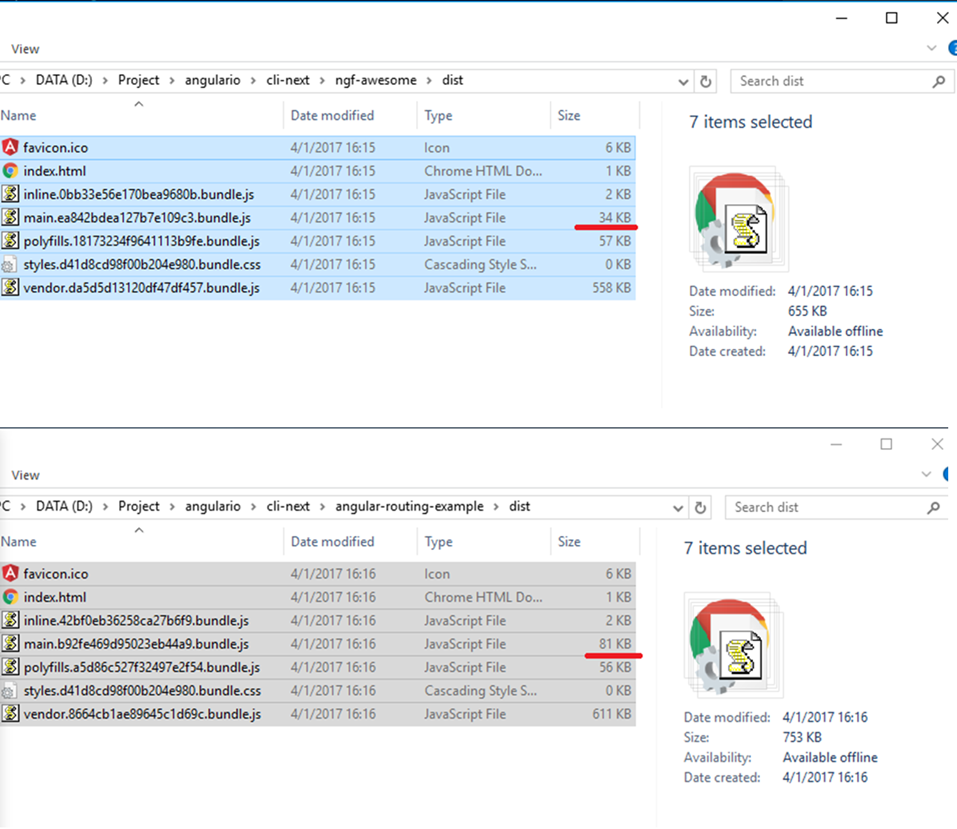
# ANGULAR 4 CÓ GÌ MỚI?

## 1. Tổng quan

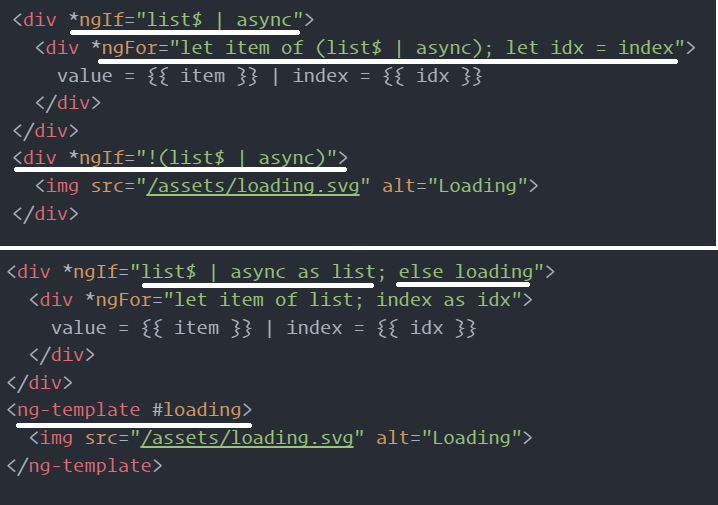
* New features: Những tính năng mới có trong phiên bản mới.
* Breaking changes: Những thay đổi ở phiên bản mới có thể làm hệ thống hiện tại bị lỗi khi nâng cấp.
* Deprecated: Những thay đổi không được sử dụng trong phiên bản mới, và ở phiên bản tiếp theo sẽ bị loại bỏ đi.

## 2. New features

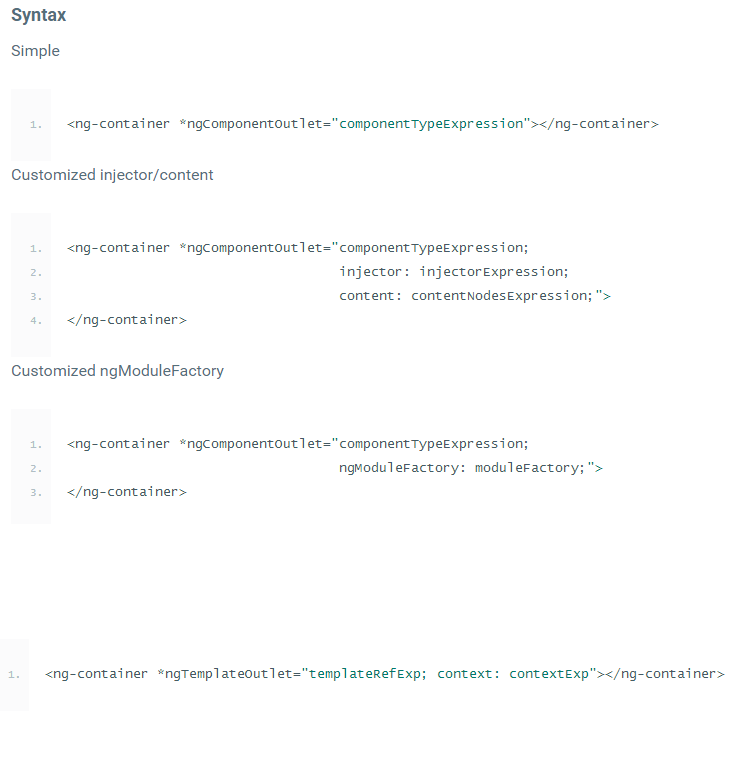
* Smaller & Faster: new View Engine giúp giảm kích thước code gen ra khoảng 60% so với trước đây.
* Tương thích với TypeScript 2.1, 2.2, Angular Universal, Source Maps for Templates.

[[](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/kawsklpflk_angular%204-1.png)](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/kawsklpflk_angular%204-1.png)

* Cải tiến ngIf, ngFor: tạo ra các local variable, if/else:

[[](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/wivpdcs4rb_angular%204-2.png)](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/wivpdcs4rb_angular%204-2.png)

* Giới thiệu NgComponentOutlet và NgTemplateOutlet tương thích với \* syntax.  
  Cú pháp của NgComponentOutlet và NgTemplateOutlet:

[[](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/4lm96kolt4_angular%204-3.png)](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/4lm96kolt4_angular%204-3.png)

Tạo dynamic component trong Angular:

[[](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/dqa2gszpn5_angular%204-4.png)](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/dqa2gszpn5_angular%204-4.png)

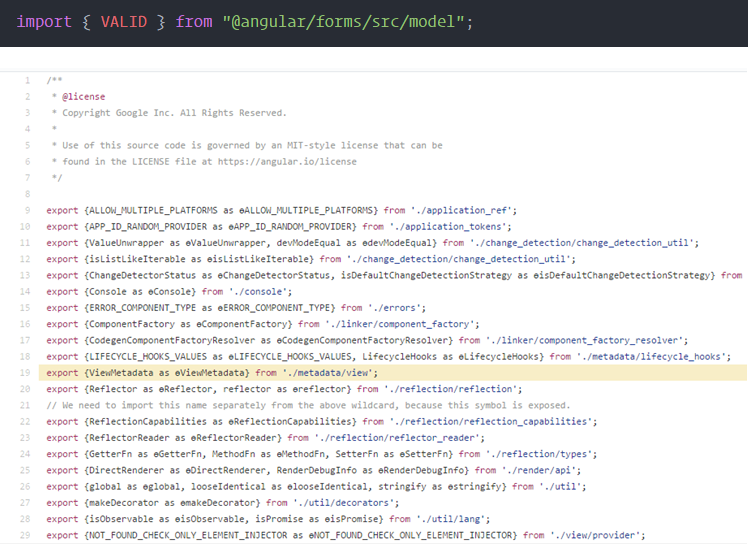
[[](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/bwi6hw5mcr_angular%204-5.png)](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/bwi6hw5mcr_angular%204-5.png)

## 3. Breaking changes

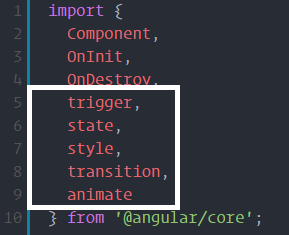
* Lifecyle events: thay thế toàn bộ là interface nên phải thay thế hết thành “implements”.
* Không cho phép deep imports và tất cả các export được đặt ký tự ɵ ở đầu thì không được phép sử dụng trong ứng dụng của bạn.

[[](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/fkncx6e2h8_angular%204-6.png)](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/fkncx6e2h8_angular%204-6.png)

Các imports không hợp lệ trong Angular 4:

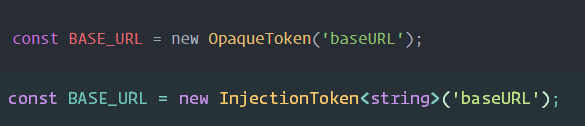
[[](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/wfac5sph0p_angular%204-7.png)](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/wfac5sph0p_angular%204-7.png)

* Animation đã tách riêng khỏi @angular/core, thay vào đó là import BrowserAnimationsModule từ @angular/platform-browser/animations và các thành phần như trigger từ @angular/animations.

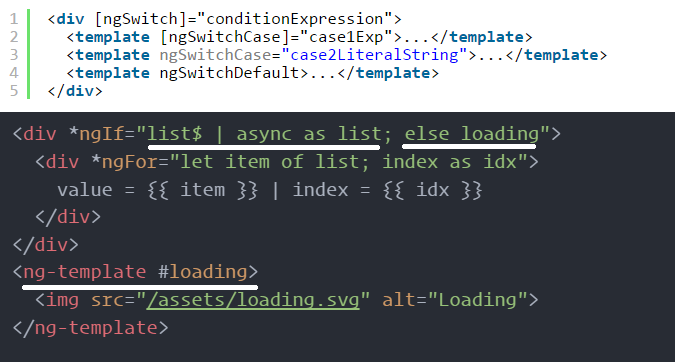
[[](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/mcayeyjawn_angular%204-8.png)](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/mcayeyjawn_angular%204-8.png)

## Deprecated

* OpaqueToken thay thế bằng InjectionToken<?>

[[](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/k72i602zcy_angular%204-9.png)](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/k72i602zcy_angular%204-9.png)

* Renderer thay thế bằng Renderer2
* <template> thay thế bằng <ng-template>

[[](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/4n3etv8pc9_angular%204-10.png)](https://s3-ap-southeast-1.amazonaws.com/kipalog.com/4n3etv8pc9_angular%204-10.png)

# XÂY DỰNG ỨNG DỤNG MINH HỌA

# TỔNG KẾT