Báo cáo giữa kì Lập trình Web

**TÌM HIỂU VỀ ANGULARJS**

1. **TỔNG QUAN VỀ ANGULARJS:**
   1. **Lịch sử phát triển:**

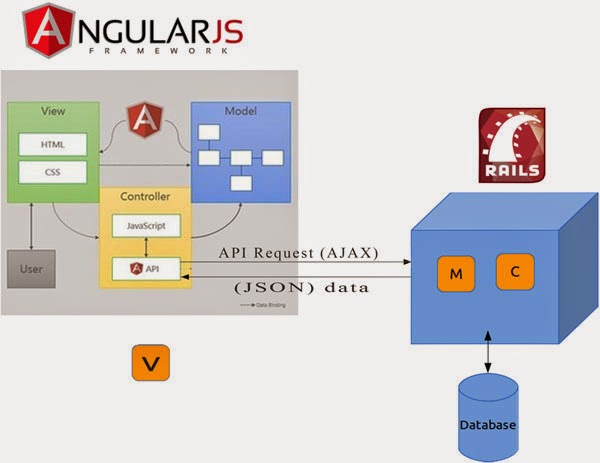
- Dự án AngularJS được bắt đầu từ năm 2009, do lập trình viên Misko Hevery tại Google viết ra. Misko và nhóm lúc này đang tham gia vào 1 dự án của Google tên là Google Feedback. Với AngularJS, Misko đã rút ngắn số dòng code front-end từ 17000 dòng còn chỉ khoảng 1500. Với sự thành công đó, đội ngũ của dự án Google Feedback quyết định phát triển AngularJS theo hướng mã nguồn mở.

- Trước năm 2000, công nghệ HTML hỗ trợ tốt cho các trang web tĩnh. Khi bạn xây dựng 1 trang web với PHP, Node/Express, hay Ruby thì nó cũng chỉ là một trang web tĩnh với nội dung được thay đổi khi bạn gửi request về máy chủ, máy chủ sẽ render 1 trang với nội dung tương ứng. Tuy nhiên mọi thứ đã thay đổi nhiều từ sự phát triển của HTML5, nhất là khi có sự chống lưng từ những ông lớn như Google, Yahoo, Facebook, và sự tập hợp đông đảo của cộng đồng mã nguồn mở. Trong đó kể đến AngularJS giúp HTML động hơn.

* 1. **Angular JS là gì ?**

- AngularJS là một framework JavaScript có cấu trúc cho các ứng dụng web động. Nó cho phép bạn sử dụng HTML như là ngôn ngữ mẫu và cho phép bạn mở rộng cú pháp của HTML để diễn đạt các thành phần ứng dụng của bạn một cách rõ ràng và súc tích. Hai tính năng cốt lõi: Data binding và Dependency injection của AngularJS loại bỏ phần lớn code mà bạn thường phải viết, giúp giảm bớt quá trình phát triển ứng dụng web. AngularJS xảy ra trong tất cả các trình duyệt, làm cho nó trở thành đối tác lý tưởng của bất kỳ công nghệ Server nào. Từ nhiều năm trước, khi HTML mới bắt đầu, nó được dự định để xây dựng trang web hoặc có thể nói đó là một cách để hiển thị tài liệu tĩnh, không để xây dựng một ứng dụng web động. Với sự ra đời của Angular, HTML lần đầu tiên đã được dùng để xây dựng các ứng dụng web động.

- Phân tích sơ đồ làm việc với AngularJS để : Web của bạn sẽ kết nối với server API, lấy về result dưới dạng JSON, Angular JS sẽ đóng vai trò data-driven để biến JSON đó thành những Models. Bạn có thể nhanh chóng update lên HTML content mà không cần phải refresh lại trang.



Nói qua cơ chế hoạt động SPA- Single-page Application, ở lần request đầu tiên, hệ thống sẽ trả về tất cả code xử lý cũng như code hiển thị của toàn bộ trang web, ở những yêu cầu tiếp theo client chỉ phải requets những phần nào mình cần, và server sẽ trả về dữ liệu dưới dạng thô, giúp rút ngắn thời gian truyền tải, giúp nâng cao trải nghiệm của người dùng hơn.

=> Angular thích hợp sử dụng nhất cho những ứng dụng có 1 trang SPA (Đó là một trang web hay một ứng dụng web, mà tất cả những thao tác xử lý của trang web đều được diễn ra trên một trang duy nhất. Với mục đích chính là nâng cao trải nghiệm người dùng tốt hơn, hiệu quả hơn, thoải mái hơn.)

Một ví dụ đơn cử dễ hiểu là trang Facebook. Nếu bạn đang lướt News feed, không cần load lại cả trang web. Bạn thấy đó, tất cả những gì thay đổi chỉ là bảng tin mới nhất được update, các khung xung quanh vẫn không đổi. Bạn sẽ cảm giác đang trượt tuyết trên một làn băng duy nhất, và không phải mất công chuyển làn băng khác.

* 1. **Các ứng dụng chung của AngularJS trong phát triển ứng dụng web:**

- AngularJS là một Framework phát triển mạnh mẽ dựa trên JavaScript để tạo các ứng dụng RICH Internet Application (RIA).

- AngularJS cung cấp cho lập trình viên những tùy chọn để viết các ứng dụng client-side trong mô hình MVC (Model View Controller) một cách rõ ràng.

- Các ứng dụng được viết bởi AngularJS tương thích với nhiều phiên bản trình duyệt web. AngularJS tự động xử lý mã JavaScript để phù hợp với mỗi trình duyệt.

- AngularJS có mã nguồn mở, miễn phí hoàn toàn, có thể tùy biến theo ý thích và được sử dụng bởi hàng ngàn lập trình viên trên thế giới. Nó hoạt động dưới giấy phép Apache License version 2.0.

Nhìn chung, AngularJS là một framework để tạo các ứng dụng lớn, các hiệu năng cao trong khi giữ cho chúng có thể dễ dàng duy trì.

* 1. **Đặc trưng của AngularJS:**

- Phát triển dự trên Javascript.

- Tạo các ứng dụng client-side theo mô hình MVC.

- Khả năng tương thích cao, tự động xử lý mã javascript để phù hợp với mỗi trình duyệt.

- Mã nguồn mở, miễn phí hoàn toàn và được sủ dụng rộng rãi.

1. **ANGULARJS FRAMEWORK:**
2. **Tại sao phải Tại sử dụng AngularJs ?**

- Angular được đưa ra bởi Google. Tại sao điều này quan trọng để biết ?

Thực tế, Google đã phát triển các tài năng và thiên tài như đội bóng của họ. Họ thực sự biết những tinh tế của trang web và những sự phát triển ứng dụng web . Ít nhất là thực tế này có thể cung cấp cho người dùng đảm bảo Angular xuất phát từ người chúng ta có thể tin tưởng .

- Ngoài ra, nếu bạn đã từng sử dụng sản phẩm của Google như Gmail hay Google Plus, bạn sẽ không ngạc nhiên với sự tương tác của chúng và cả ajax gửi liên tục mọi nơi mà không phải làm mới toàn bộ trang web và để sử dụng.Hãy để ý, với Google, khi bạn nhập những gì muốn tìm kiếm, tất cả những thay đổi chỉ là những kết quả được xuất ra, riêng khung Google bên trên vẫn đứng yên dù bạn có chuyển sang trang kết quả tìm kiếm 2, 3,v.v..

1. **AngularJs được lập trình như thế nào ?**
2. **Cài đặt AngularJS:**

* **Tải AngularJS:**

Truy cập vào trang web https://angularjs.org/ và tải về bản angularjs mới nhất. Bản hiện mới nhất hiện tại là 1.6.9

* **Chèn Angular vào ứng dụng:**

*Tự host AngularJS:*

<script src="angular.min.js"></script>

Theo cách này thì cần phải download angularjs về máy và nhúng trực tiếp vào ứng dụng.

*Dùng phiên bản có sẵn trên server của Google:*

Ngoài cách trên ra bạn cũng có thể sử dụng phiên bản nén của AngularJS có sẵn trên server của Google. Sử dụng cách này có 2 điều lợi là tiết kiệm băng thông cho trang web của bạn và AngularJS sẽ được load nhanh hơn nếu máy của người dùng đã cache AngularJS.

<script src=”https://ajax.googleapis.com/ajax/libs/angularjs/1.6.3/angular.min.js”></script>

Nhưng cách này không hoạt động nếu không được kết nối mạng.

1. **Ví dụ đơn giản:**

Các bước tạo một ứng dụng AngularJS

**Bước 1:** Tải framework theo 1 trong hai cách đã nêu

Là một khung JavaScript thuần, thư viện AngularJS có thể được thêm vào bởi sử dụng thẻ <script>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

**Bước 2:** Định nghĩa ứng dụng AngularJS sử dụng ng-app directive.

<div ng-app=””>

...

</div>

**Bước 3:** Định nghĩa tên một model sử dụng ng-model directive.

<p>Nhập tên của bạn: <input type="text" size="30" ng-model="ten"></p>

**Bước 4:** Gắn kết giá trị của model đã được định nghĩa ở trên sử dụng ngbind directive.

<p>Nhập tên của bạn: <input type="text" size="30" ng-model="ten"></p>

<p>Xin chào <span ng-bind="ten"></span>!</p>

**Các bước để chạy một ứng dụng AngularJS**

**Sử dụng các bước bên trên để tạo ra một trang HTML.**

# VD1: Ứng dụng HelloWorld với AngularJS:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Tạo ứng dụng AngularJS đầu tiên</title>

<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.6.3/angular.min.js"></script>

</head>

<body>

<h1>Ứng dụng HelloWorld với AngularJS </h1>

<div ng-app="">

<p>Nhập tên của bạn: <input type="text" size="30" ng-model="ten"></p>

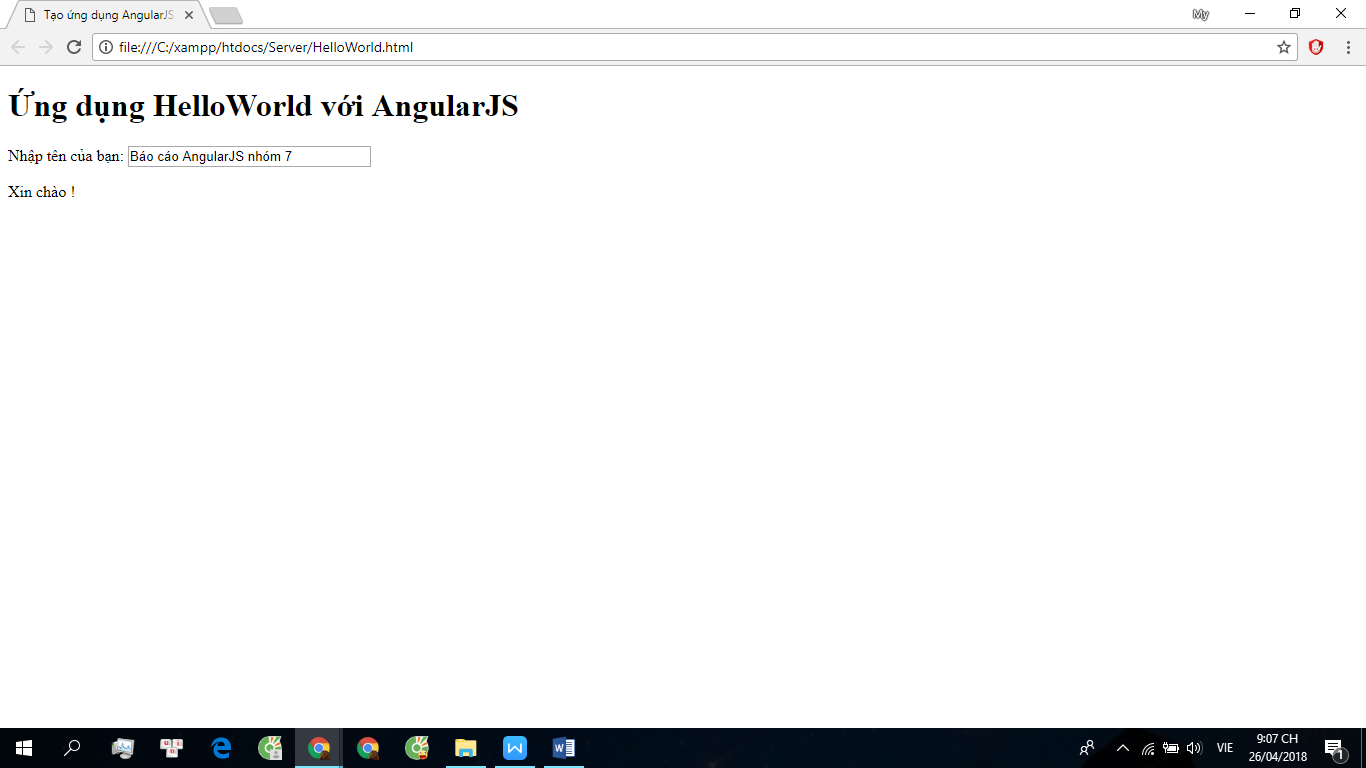
<p>Xin chào <span ng-bind="ten"></span>!</p>

</div>

</body>

</html>

**Kết quả:** Mở trang HelloWorld.html trên trình duyệt, nhập tên và xem kết quả.



# VD2: Hiệu ứng Animation với AngularJS:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Hiệu ứng div</title>

<style>

div {

transition: all linear 0.5s;

background-color: lightblue;

height: 100px;

width: 100%;

position: relative;

top: 0;

left: 0;

}

.ng-hide {

height: 0;

width: 0;

background-color: transparent;

top:-200px;

left: 200px;

}

</style>

</head>

<body ng-app="ngAnimate">

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

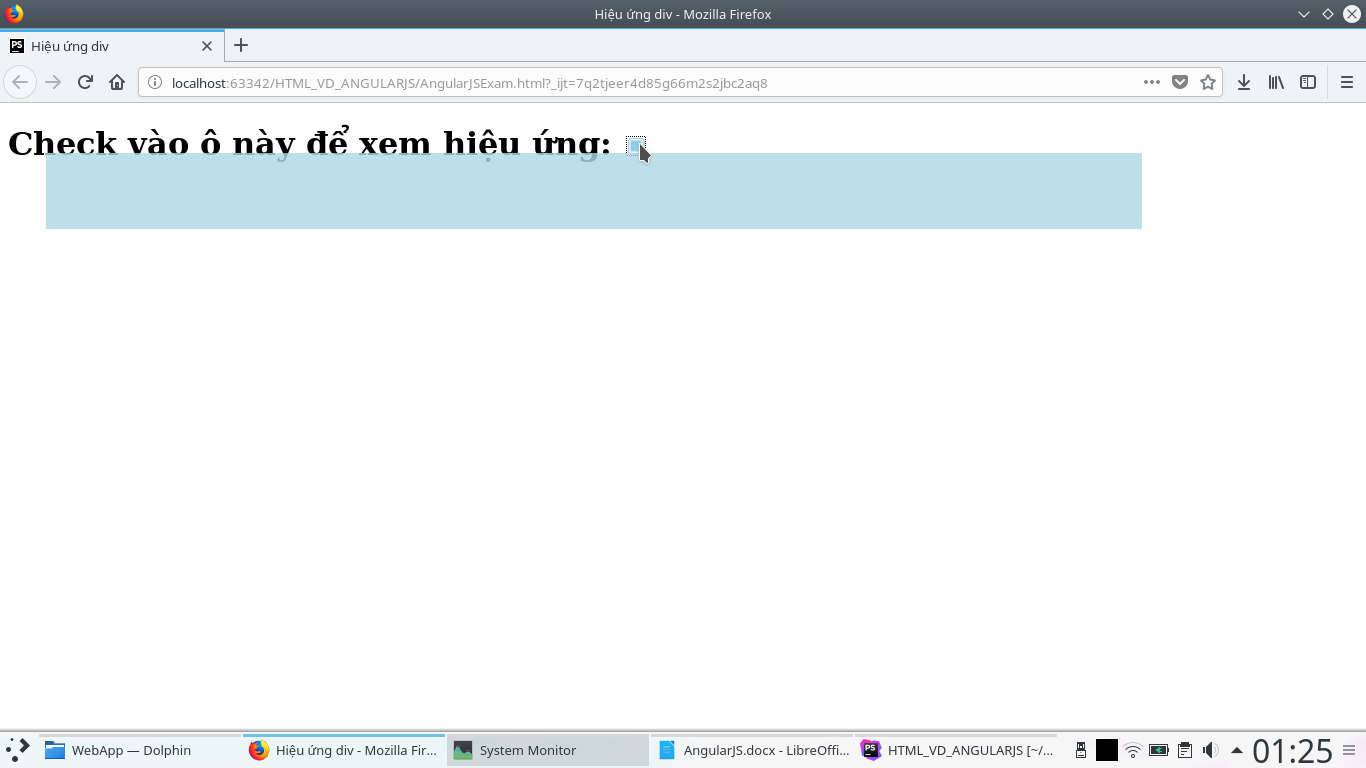
<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular-animate.js"></script>

<h1>Check vào ô này để xem hiệu ứng: <input type="checkbox" ng-model="myCheck"></h1>

<div ng-hide="myCheck"></div>

</body>

</html>

**Kết quả:** Click chuột vào ô checkbox và xem hiệu ứng xuất hiện.

# VD3: HTML DOM với AngularJS:

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>Ví dụ về AngularJS HTML DOM</title>

<script src="https://ajax.googleapis.com/ajax/libs/angularjs/1.6.9/angular.min.js"></script>

</head>

<body>

<h1>Ví dụ HTML DOM</h1>

<div ng-app="Demo" ng-init="sw=true" ng-controller="Ctrlr">

<p>

<button ng-disabled="sw">Bấm vào đây!</button>

</p>

<p>

<input type="checkbox" ng-model="sw" ng-lick="chg(sw)"/>{{textchg}}

</p>

<p>

{{ sw }}

</p>

<script>

**var** app = angular.module('Demo', []);

app.controller('Ctrlr', **function**($scope) {

$scope.textchg = "Click để hiện lên";

$scope.chg = **function**(sw) {

**if**(sw)

$scope.textchg = "Click để hiện lại";

**else**

$scope.textchg = "Click để vô hiệu hoá";

}

});

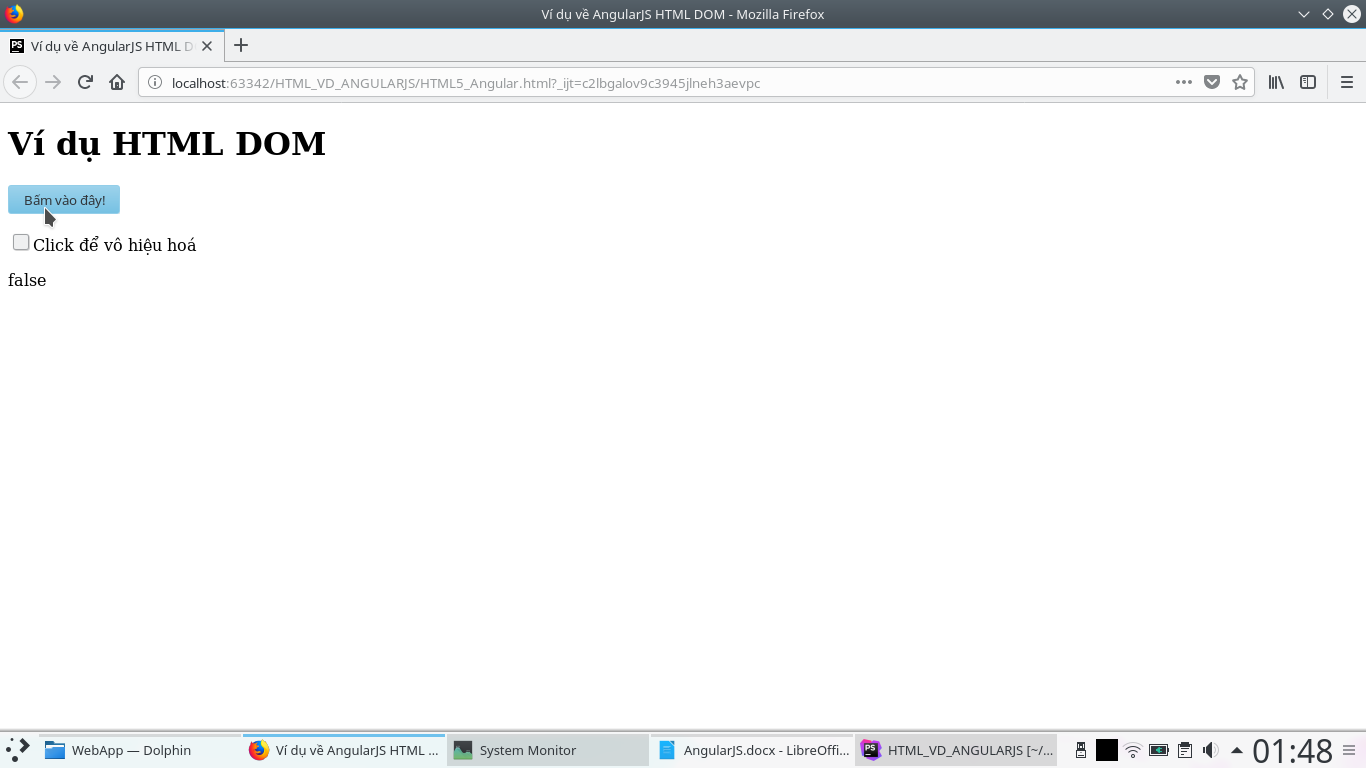
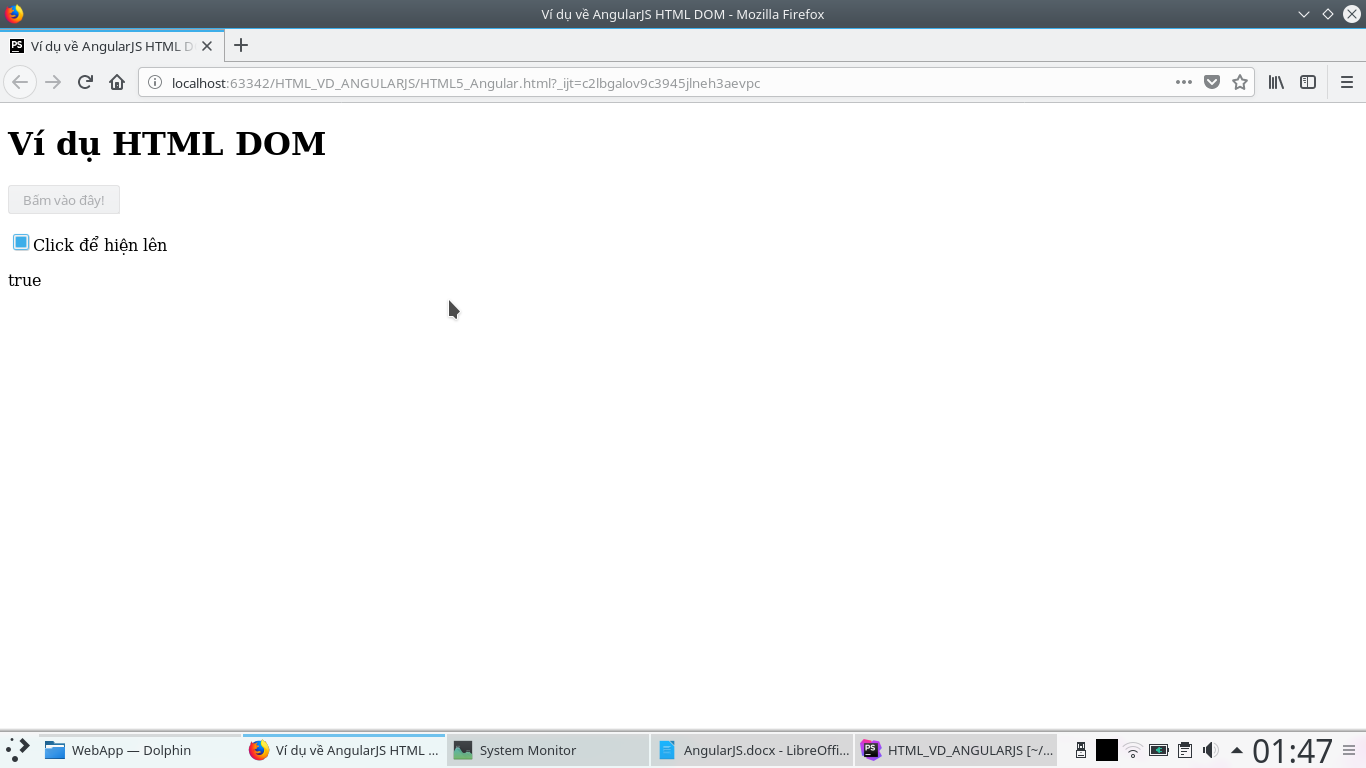
</script>

</div>

</body>

</html>

**Kết quả:** Ban đầu nút bấm vô hiệu hoá {{sw}} hiện giá trị true (mặc định), click chuột vào ô checkbox để uncheck nút bấm hiện lên, {{sw}} trả về giá trị false hiện ra, có thể click được nút bấm. Ngược lại, khi re-check lại, nút bấm mờ đi và vô hiệu hoá, không thể click chuột.



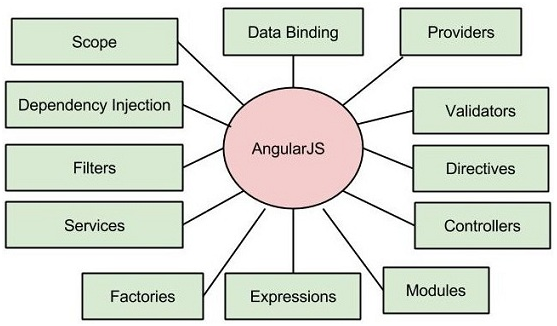
Ban đầu

Lúc sau

1. **Các tính năng cốt lõi của AngularJS:**

Angular là một MVW Javascript framework rất hấp dẫn để xây dựng nên SPA và nó cung cấp những tính năng rất hiện đại cho việc xây dựng web. MVW là viết tắt của Model-View-Whatever, bạn có thể thấy rằng tham vọng cũng như những gì nó mang lại là rất lớn, và nó sẽ hỗ trợ chúng ta một cách mềm dẻo để phát triển mọi thể loại ứng dụng. Trong MVW, chữ W có thể được thay thế để bạn có thể đi theo các hướng MVC (ModelView-Controller) hoặc MVVM (Model-View-ViewModel ).

Nhìn tổng quát ở sơ đồ bên dưới, ta có thể thấy được những tính năng chính do Angular JS cung cấp :



- Data-binding sự gắn kết data lưu trong model với view : Angular có khả năng tự động đồng bộ hoá data giữa Model và View. Data được thay đổi trên View (thông qua input field) sẽ ngay lập tức được cập nhật vào Model javascript object. Ngược lại, javascript object khi có thay đổi, có thể là từ JSON phía server API trả về, sẽ được tự động cập nhật hiển thị lên HTML view ngay mà bạn không cần phải can thiệp vào. Đây chắc chắn là thế mạnh mà bạn sẽ cực kỳ thích thú.

- Scope: Scope có thể hiểu là những objects model ở tầng application (chú ý, nó không chỉ là Model để lưu các giá trị thuộc tính của object trong OOP, mà nó còn lưu cả các cài đặt cho application ở thời điểm hiện tại nữa). Và chúng sẽ đóng vai trò kết nối giữa controller với view. Chẳng hạn, bạn sẽ gán action cho button đến một hàm trong scope.

-Controller: Controller trong Angular JS sẽ chứa các javascript method, và giới hạn trong một scope cụ thể.

-Services: AngularJS cung cấp một vài services sẵn, ví dụ như $http để thực hiện một lời gọi AJAX (XMLHttpRequest) đến server API. Những services này là những object duy nhất, dùng chung trong toàn app, và nó chỉ được khởi tạo một lần. Bạn sẽ có thể sẵn sàng sử dụng các object này ngay bất cứ lúc nào, và Angular framework sẽ quản lý những object này giúp bạn. Ví dụ: $http, $controller, $document, $compile, $rootElement , $rootScope …

-Filters - các bộ lọc: Bạn sẽ sử dụng các bộ lọc này để lọc ra được những item từ một arrays nào đó.

-Directives - điều hướng: Các điều hướng là những đánh dấu (markers) trên các thành phần DOM. Nó có thể là bản thân cả element, thuộc tính, css,... để giúp bạn thao tác với element đó. Directives cũng có thể được sử dụng để tạo ra những thẻ HTML tuỳ biến, các widgets. AngularJS cung cấp sẵn một số directive thường dùng, mà sau này bạn sẽ quen thuộc, ví dụ như: ngBind, ngModel, …

-Templates - mẫu giao diện: Template sẽ những view hiển thị các thông tin từ model do controller chỉ thị. Nó có thể là một file (ví dụ index.html), có thể là nhiều view như là những thành phần con (partials) để ghép lại thành một màn hình.

-Routing - chuyển hướng trang web: Nếu bạn đã từng làm việc với các framework MVC, bạn sẽ hiểu khái niệm này. Routing sẽ giúp bạn chỉ định view nào sẽ hiển thị, nếu người dùng truy cập vào URL có patterns như thế nào. Nghĩa là nó sẽ thay đổi views hiển thị tương ứng với ngữ cảnh.

-MVM - model view whatever: MVW là một design pattern mà chia một ứng dụng thành các phần Model, View, và thường là Controller tương tự là MVC truyền thống. Mỗi phần sẽ có khả năng và vai trog đáp ứng riêng. So với MVC truyền thống, MVW còn pha trộn thêm MVVW (Model-ViewViewModel). Và cái tên MVW Model View Whatever là do Angular JS team đặt ra.

-Deep linking - liên kết sâu: Deep linking liên kết sâu cho phép ta mã hoá tình trạng của ứng dụng vào URL, và ta có thể bookmark lại được. Sau đó, ứng dụng có được dựng lại hoàn toàn trạng thái của nó sau này khi ta access vào URL đó.

-**Dependency Injection**: AngularJS có sẵn một hệ thống **Injection** để developers có thể dễ dàng tạo, hiểu, và kiểm tra ứng dụng.

1. **Các thành phần quan trọng của AngularJS Framework:**

AngularJS framework có thể được có thành ba phần quan trọng sau:

- ng-app: directive này định nghĩa và liên kết một ứng dụng AngularJS tới HTML.

- ng-model: directive này gắn kết giá trị của dữ liệu ứng dụng AngularJS đến các điều khiển đầu vào HTML.

- ng-bind: directive này gắn kết dữ liệu ứng dụng AngularJS đến các thẻ HTML.

1. **ƯU ĐIỂM VÀ NHƯỢC ĐIỂM CỦA ANGULARJS:**
2. **Ưu điểm:**

- Với concept ý tưởng, cũng như những thành phần có sẵn mà Angular cung cấp, ta có thể thấy nó giúp phát triểnnhững ứng dụng SPA (Single Page Application) hiệu quả và tiết kiệm về băng thông, cũng như cho trải nghiệm người dùng tốt hơn, việc này sẽ rất được thực hiện một các rõ ràng , và dễ cho bảo trì code.

- Với việc mà binding được data vào HTML DOM, nó rất nhanh update data qua lại giữa view và model do đó giúp người dùng cảm giác linh hoạt, thân thiện.

- Code viết với AngularJS rất dễ test.

- AngularJS sử dụng injection để bạn ngăn cách và hiểu một cách tách bạch được những thứ liên quan đến nhau.

- Có thể viết theo hướng tái sử dụng.

- Với AngularJS, các lập trình viên sẽ chỉ cần code ít hơn cho những tính năng lớn.

- Trong AngularJS, các views là những trang HTML code thuần; các Controller thì được viết bằng javascript để thực hiện các quá trình xử lí nghiệp vụ.

- Và điều quan trọng nhất, ứng dụng viết bằng AngularJS sẽ chạy trên hầu hết các trình duyệt hiện nay, bao gồm cả các trình duyệt trên smart phones, tablets máy tính bảng, đương nhiên là bao gồm cả iOS và Android.

**2. Nhược điểm của AngularJS:**

Mặc dù có nhiều lợi ích khi sử dụng AngularJS, tuy nhiên không có gì là hoàn hảo, cần xem xét một số nhược điểm của Angular để có kế hoạch cho mình khi phát triển ứng dụng bằng AngularJS.

- Không an toàn: Vì nó là nền tảng dựa trên javaScript, có nghĩa là nó chạy phía client. Điều này đồng nghĩa với việc nếu người dùng có hiểu biết về coding, có mục đích mờ ám, thì họ có thể sửa đổi code phía client trước khi submit lên server. Do đó, một số các tác vụ quan trọng, bạn nên thực hiện phía server. Ví dụ: Authenticate và Authorization (xác thực và kiểm tra quyền của người dùng); thực hiện validation với những data được submit lên API trước khi save hoặc thao tác với data do người dùng submit lên đó. Do đó cách tốt nhất là dùng AngularJS để query vào các WebService API, không nên save trực tiếp data vào database mà không check.

- Nếu người dùng ứng dụng, bật chức năng vô hiệu hóa Javascript, bạn sẽ không thể làm gì được, webpage sẽ chỉ hiển thị được những thông tin cơ bản, và không có gì hiển thị thêm đương nhiên là các thao tác xữ lí sẽ không hoạt động. Tuy nhiên điều này có lẽ ít xảy ra. Điều may mắn là những nhược điểm này có thể khắc phục hoặc hạn chế được, đủ để ta yên tâm coding app bằng AngularJS.

1. **SỰ RA ĐỜI CỦA ANGULAR 2.0:**

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.

Những gì thay đổi?AngularJS 2.0 sẽ không còn sử dụng hoặc thay đổi đáng kể nhiều thứ ở Angular 1.x, điển hình là :

- Controllers - Kiến trúc MVC truyền thống sẽ được thay thế với một kiến trúc các component linh hoạt. Controllers là không còn cần thiết và thay thế bằng một component với component chuyên dụng được điều khiển.

- Directives - Directive Definition Object hiện tại sẽ được gỡ bỏ. Sẽ có 3 loại directive (component, decorator, template).  
- $scope - scopes và scope inheritanc được đơn giản hóa với yêu cầu phải inject $ scope được gỡ bỏ.

- angular.module - module trong angular được thay thế bằng các module có sắn của ES6 .

- jQLite -. được loại bỏ chủ yếu nhằm tăng performance.

“AngularJS 1.x is built for current browsers while  
”Angular 2.0 is being built for the browsers of the future.  
– Igor Minaz, AngularJS team