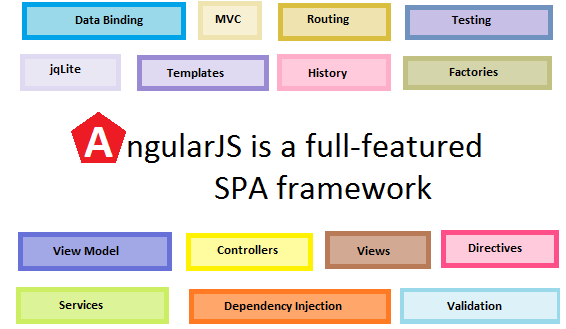
**Báo cáo AngularJS**

AngularJS là framework có cấu trúc cho các ứng dụng web động. Hai tính năng cốt lõi. Data binding và Dependency injection của AngularJS loại bỏ phần lớn code mà bạn thường phải viết. Được phát triển từ năm 2009, hiện tại được duy trì bởi google và đã ra mắt phiên bản 2.0.



Đặc trưng:

* 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 vứi mỗi trình duyệt.
* Mã nguồn mở, miễn phí hoàn toàn và được sủ dụng rộng rãi.

Tính năng cơ bản:

* Scope : là đối tượng có nhiệm vụ giao tiếp giữa controller và view của ứng dụng.
* Controller : xử lí dữ liệu cho đối tượng $scope, từ đây bên views sẽ sử dụng các dữ liệu trong scope để hiển thị ra tương ứng.
* Data-binding : tự động đồng bộ dữ liệu giữa model và view
* Service : là singleton object được khởi tạo 1 lần duy nhất cho mỗi ứng dụng, cung cấp các phương thức lưu trữ dữ liệu có sãn. ($http, $httpBackend, $sce, $controller, $document, $compile, $parse, $rootElement, $rootScope …..)
* Filter : Lọc các tập con từ tập item trong các mảng và trả về các mảng mới.
* Directive : dùng để tạo các thẻ HTML riêng phục vụ những mục đích riêng. AngularJS có những directive có sẵn như
* Ng-app: Khai báo phần tử gốc của một ứng dụng AngularJS, theo đó các directives có thể được sử dụng để khai báo ràng buộc và xác định hành vi.
* Ng-bind: Đặt văn bản của một phần tử DOM vào giá trị của một biểu thức. Ví dụ: <span ng-bind = "name"> </ span> hiển thị giá trị của 'name' bên trong phần tử span. Bất kỳ thay đổi nào đối với biến 'name' trong phạm vi của ứng dụng phản ánh ngay lập tức trong DOM.
* Ng-model: Tương tự với ng-bind, nhưng thiết lập một ràng buộc dữ liệu hai chiều giữa view và scope.
* Ng-model-options: Cung cấp điều chỉnh để cập nhật model được thực hiện như thế nào.
* Ng-class: Áp dụng một cách có điều kiện một lớp, tùy thuộc vào giá trị của một biểu thức boolean.
* Ng-controller: Chỉ định một lớp điều khiển JavaScript để đánh giá các biểu thức HTML.
* Ng-repeat: Khởi tạo một yếu tố một lần cho mỗi mục từ bộ sưu tập.
* Ng-show & ng-hide: Hiển thị hoặc ẩn một phần tử theo điều kiện, tùy thuộc vào giá trị biểu thức boolean. Hiển thị và ẩn là đạt được bằng cách thiết lập kiểu hiển thị CSS.
* Ng-switch: Tạo điều kiện cho một mẫu từ một tập hợp các lựa chọn, tùy thuộc vào giá trị của một biểu thức lựa chọn.
* Ng-view: Chỉ thị cơ sở có trách nhiệm xử lý các tuyến đường [12] giải quyết JSON trước khi rendering các khuôn mẫu do các bộ điều khiển quy định.
* Ng-if: Chỉ thị nếu tuyên bố rằng instantiates các yếu tố sau đây nếu điều kiện là đúng sự thật. Khi điều kiện là sai, phần tử sẽ bị xóa khỏi DOM. Khi đúng sự thật, một bản sao của phần tử được biên dịch sẽ được chèn lại.
* Ng-aria: Một mô-đun hỗ trợ khả năng truy cập của các thuộc tính ARIA thông thường.
* Ng-animate: Một mô-đun cung cấp hỗ trợ chuyển đổi JavaScript, CSS3 và các móc hoạt hình khung hình CSS3 trong các chỉ thị lõi và tùy chỉnh hiện có.
* Temple : một thành phần của view, hiển thị thông tin từ controller
* Routing : chuyển đổi giữa các action trong controller, qua lại giữa các view.
* MVC & MVVM : mô hình thiết kế để phân chia các ứng dụng thành nhiều phần khác nhau (gọi là Model, View và Controller) mỗi phần có một nhiệm vụ nhất định. AngularJS không triển khai MVC theo cách truyền thống, mà gắn liền hơn với Model-View-ViewModel.
* Deep link : Liên kết sâu, cho phép bạn mã hóa trạng thái của ứng dụng trong các URL để nó có thể bookmark với công cụ tìm kiếm. Các ứng dụng có thể được phục hồi lại từ các địa chỉ URL với cùng một trạng thái.
* Dependency Injection: AngularJS có sẵn một hệ thống con dependency injection để giúp các lập trình viên tạo ra các ứng dụng dễ phát triển, dễ hiểu và kiểm tra.

Các components chính:

* ng-app : định nghĩa và liên kết một ứng dụng AngularJS tới HTML.
* ng-model : gắn kết giá trị của dữ liệu ứng dụng AngularJS đến các điều khiển đầu vào HTML.
* ng-bind : gắn kết dữ liệu ứng dụng AngularJS đến các thẻ HTML.