Angular JS and Its Important Component

BIBHISHAN SUTAR [1], DR. PRATIBHA ADKAR [2]
[1, 2] *P.E.S Modern College Of Engineering, Pune*

Abstract -- To develop light and small web applications that are easy to create and simple to test and maintain when they are extend their code. This experienced guide introduces Angular JS, release by the search engine Google and uses the MVC architecture with the onset up source JavaScript. Angular JS is highly enriched in prominent attribute for designing the client side applications with many features and properties. This research paper helps the user to understand what is Angular JS, why Angular JS, MVC Architecture of Angular JS. This report also explains the various important components of Angular JS. [2]

Indexed Terms: MVC, JS, DOM, Angular JS

I. INTRODUCTION

➢ What is Angular JS?

Angular JS is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly. Angular JS's data binding and dependency injection eliminate much of the code you would otherwise have to write. And it all happens within the browser, making it an ideal partner with any server technology. [1]

Angular JS is what HTML would have been, had it been designed for applications. HTML is a great declarative language for static documents. It does not contain much in the way of creating applications, and as a result building web applications is an exercise in what do I have to do to trick the browser into doing what I want?[1]

The impedance mismatch between dynamic applications and static documents is often solved with:

- A library: a collection of functions which are useful when writing web apps. Your code is in charge and it calls into the library when it sees fit. E.g., jQuery.
- Frameworks: A particular implementation of a web application, where your code fills in the details. The framework is in charge and it calls into your code when it needs something app specific. E.g., durandal, ember, etc. [2]

II. Why Angular JS?

Angular is the only framework that doesn't make MVC seem like putting lipstick on a pig.

Most frameworks nowadays are simply a bundling of existing tools. They are an integrated tool set, but not very elegant. Angular is the next generation framework where each tool was designed to work with every other tool in an interconnected way.

Here are 5 reasons why you should be using Angular today. [4]

1. MVC done right:

Most frameworks implement MVC by asking you to split your app into MVC components, then require you to write code to string them up together again. That's a lot of work. Angular implements MVC by asking you to split your app into MVC components, then just let Angular do the rest. Angular manages your components for you and also serves as the pipeline that connects them. [1]

Because Angular acts as the mediator, developers also won't feel tempted to write shortcuts between components that break abstractions just to make them fit easier. [3]

2. A declarative user interface:

Angular uses HTML to define the app's user interface. HTML is a declarative language which is more intuitive and less convoluted than defining the interface procedurally in JavaScript. HTML is also less brittle to reorganize than an interface written in

JavaScript, meaning things are less likely to break. Plus you can bring in many more UI developers when the view is written in HTML. [6]

HTML is also used to determine the execution of the app. Special attributes in the HTML determine which controllers to use for each element. These attributes determine "what" gets loaded, but not "how". This declarative approach greatly simplifies app development in a sort of WYSIWYG (what you see is what you get) way. Rather than spending time on how the program flows and what should get loaded first, you simply define what you want and Angular will take care of the dependencies. [3]

3. Data models are POJO:

Data models in Angular are plain old JavaScript objects (POJO) and don't require extraneous getter and setter functions. You can add and change properties directly on it and loop over objects and arrays at will. Your code will look much cleaner and more intuitive, the way mother nature intended. [2]

Traditional data models are the gatekeepers of data and are responsible for data persistence and server syncing. Those data models behave like smart data providers. But since Angular's data models are plain objects, they behave more like a cork board. A cork board is nothing more than a temporary storage area for people to put and retrieve data. However, Angular's cork boards work closely with a controller and view. To differentiate it from the traditional sense of data models, Angular calls them "scopes".

All properties found on the scope object are automatically bound to the view by Angular. Meaning, Angular quietly watches for changes to these properties and updates the view automatically. [5]

The scope has no data to begin with and relies on the controller to feed it data according to business logic needs.

4. Service providers where they belong:

Services are exactly what they sound like. They don't get involved with the MVC of your app, but simply provide an outward API to expose whatever you want it to expose. Most of the time it syncs up to a server to maintain an offline data store and exposes methods to push and pull data to and from a server. Or it can be used to create a resource sharing service that allows multiple controllers to share the same resources. [5]

Services are designed to be standalone objects separate from your app and allow your controller to be remain lean and dedicated to the view and scope that it is assigned to. Of course, implementing services is not required and it is perfectly acceptable to do some light lifting inside your controller to avoid over complexity. [4]

5. Unit testing ready:

What description of Angular would be complete without talking about its unit testing readiness? The whole of Angular is linked together by Dependency Injection (DI). It's what it uses to manage your controllers and scopes. Because all your controllers depend on DI to pass it information, Angular's unit tests are able to usurp DI to perform unit testing by injecting mock data into your controller and measuring the output and behavior. In fact, Angular already has a mock HTTP provider to inject fake server responses into controllers. [1]

This beats the more traditional way of testing web apps by creating individual test pages that invoke one component and then interacting with it to see if it works.

These 5 reasons should give you an idea of why Angular JS is so powerful. Not all web apps should use Angular JS. For example, if you are writing a game or a computationally intensive math program, there is no reason why Angular would fit your particular problem domain. But for generic web apps, it should serve as a viable framework to build upon. [6]

III. ANGULAR JS - MVC ARCHITECTURE

Model **V**iew **C**ontroller or MVC as it is popularly called, is a software design pattern for developing web applications. A Model View Controller pattern is made up of the following three parts

- Model − It is the lowest level of the pattern responsible for maintaining data.

- View − It is responsible for displaying all or a portion of the data to the user.

- Controller − It is a software Code that controls the interactions between the Model and View. [1]

MVC is popular because it isolates the application logic from the user interface layer and supports separation of concerns. The controller receives all requests for the application and then works with the model to prepare any data needed by the view. The view then uses the data prepared by the controller to generate a final presentable response. The MVC abstraction can be graphically represented as follows. [2]

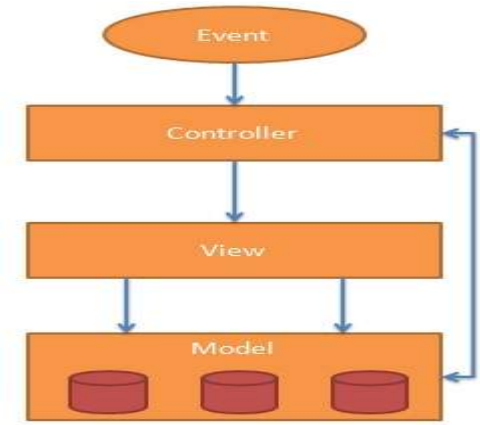

Fig. 1: Angular JS - MVC Architecture [5]

1. The Model:
The model is responsible for managing application data. It responds to the request from view and to the instructions from controller to update itself. [2]

2. The View:
A presentation of data in a particular format, triggered by the controller's decision to present the data. They are script-based template systems such as JSP, ASP, PHP and very easy to integrate with AJAX technology. [2]

3. The Controller:
The controller responds to user input and performs interactions on the data model objects. The controller receives input, validates it, and then performs business operations that modify the state of the data model.

Angular JS is a MVC based framework. [3]

IV. IMPORTANT COMPONENT OF ANGULAR JS

The diagram shows the various Component of Angular JS.

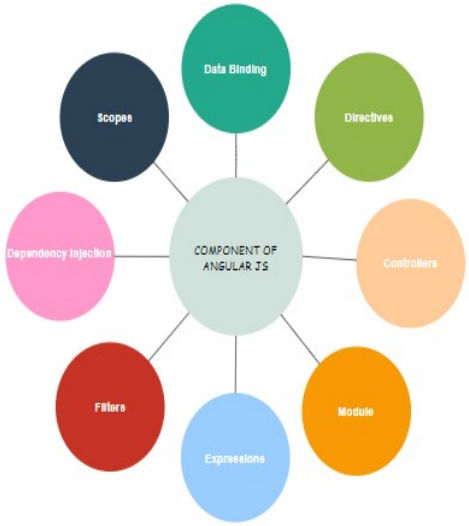

Fig. 2: Important Component of Angular JS [6]

1. Angular JS Data Binding:
Data binding is a very useful and powerful feature used in software development technologies. It acts as a bridge between the view and business logic of the application. [5]

Angular JS follows Two-Way data binding model.

- One-Way Data Binding:

The one-way data binding is an approach where a value is taken from the data model and inserted into an HTML element. There is no way to update model from view. It is used in classical template systems. These systems bind data in only one direction.

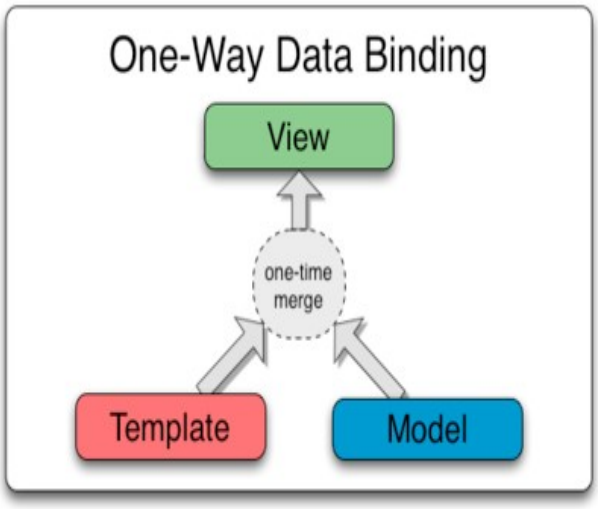

Fig. 3: One-Way Data Binding [1]

- Two-Way Data Binding:

Data-binding in Angular apps is the automatic synchronization of data between the model and view components.

Data binding lets you treat the model as the single-source-of-truth in your application. The view is a projection of the model at all times. If the model is changed, the view reflects the change and vice versa.

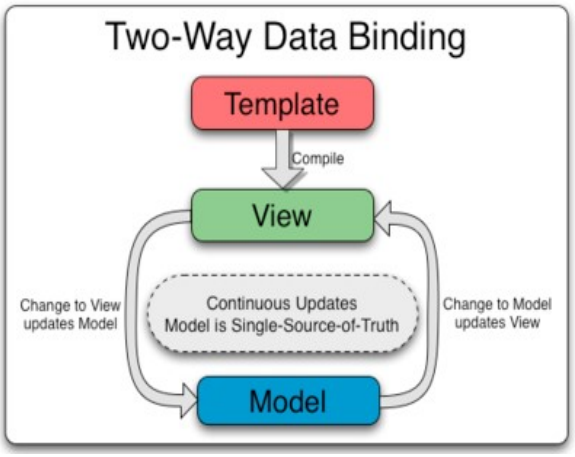

Fig. 4: Two-Way Data Binding [1]

2. Angular JS Directives:

Angular JS facilitates you to extend HTML with new attributes. These attributes are called directives.

There is a set of built-in directive in Angular JS which offers functionality to your applications. You can also define your own directives.

Directives are special attributes starting with ng- prefix. Following are the most common directives:

- ng-app: This directive starts an Angular JS Application.
- ng-init: This directive initializes application data.
- ng-model: This directive defines the model that is variable to be used in Angular JS.
- ng-repeat: This directive repeats html elements for each item in a collection. [1]

- ng-app directive:

ng-app directive defines the root element. It starts an Angular JS Application and automatically initializes or bootstraps the application when web page containing Angular JS Application is loaded. It is also used to load various Angular JS modules in Angular JS Application. [2]

- ng-init directive:

ng-init directive initializes an Angular JS Application data. It defines the initial values for an Angular JS application.

- ng-model directive:

ng-model directive defines the model/variable to be used in Angular JS Application.

- ng-repeat directive:

ng-repeat directive repeats html elements for each item in a collection. In following example, we've iterated over array of countries. [1]

3. Angular JS Controllers:

Angular JS controllers are used to control the flow of data of Angular JS application. A controller is defined using ng-controller directive. A controller is a JavaScript object containing attributes/properties and functions. Each controller accepts $scope as a parameter which refers to the application/module that controller is to control. [1]

4. Angular JS Module:

In Angular JS, a module defines an application. It is a container for the different parts of your application like controller, services, filters, directives etc.

A module is used as a Main() method. Controller always belongs to a module.

5. Angular JS Expressions:

In Angular JS, expressions are used to bind application data to HTML. Angular JS resolves the expression, and return the result exactly where the expression is written. [2]

Expressions are written inside double braces {{expression}}.They can also be written inside a directive: [3]

ng-bind="expression".

Angular JS expressions are very similar to JavaScript expressions. They can contain literals, operators, and variables. [1]

6. Angular JS Filters:

In Angular JS, filters are used to format data. Following is a list of filters used for transforming data.

You can add filters to expressions by using the pipe character |, followed by a filter. [1]

7. Angular JS Dependency Injection:

Angular JS comes with a built-in dependency injection mechanism. It facilitates you to divide your application into multiple different types of components which can be injected into each other as dependencies.

Dependency Injection is a software design pattern that specifies how components get holds of their dependencies. In this pattern, components are given their dependencies instead of coding them within the component.

Modularizing your application makes it easier to reuse, configure and test the components in your application. Following are the core types of objects and components: [5]

- value
- factory
- service
- provider
- constant

These objects and components can be injected into each other using Angular JS Dependency Injection.

- Value:

In Angular JS, value is a simple object. It can be a number, string or JavaScript object. It is used to pass values in factories, services or controllers during run and config phase. [3]

- Injecting a value:

To inject a value into Angular JS controller function, add a parameter with the same when the value is defined. [4]

- Factory:

Factory is a function that is used to return value. When a service or controller needs a value injected from the factory, it creates the value on demand. It normally uses a factory function to calculate and return the value.

- Injecting values into factory:

To inject a value into Angular JS controller function, add a parameter with the same when the value is defined. [1]

8. Angular JS Scopes:

The Scope is an object that is specified as a binding part between the HTML (view) and the JavaScript (controller). It plays a role of joining controller with the views. It is available for both the view and the controller. [1]

V. CONCLUSIONS

Angular JS can do everything that jQuery does and much more, yet is roughly equivalent in download size. It is easy to both write and run unit tests and end-to-end tests for Angular JS applications. Dependency management is effortless and intuitive. Binding dynamic data to your views is straightforward and powerful. Directives, routing,

services, validation, resources, animation and localization - are equally thought out and useful tools. Angular JS is a solid foundation for building testable web applications that scale.

This research paper helps the user to understand what is Angular JS, why Angular JS, MVC Architecture of Angular JS. This report also explains the various important components of Angular JS. [3]

REFERENCES

[1] Preeti Yadav "A comparative study of versions of Java Script" ISSN 0973-1873 Volume 13, Number 8 (2017), pp. 2065-2073.

[2] Anurag Kumar and Ravi Kumar Singh" Comparative analysis of Angular Js and React Js "Vol.(7)Issue(4), pp.225-227

[3] Madhuri A. Jadhav et al, "Single Page Application using Angular JS", International Journal of Computer

[4] Madhuri A. Jadhav et al, "Single Page Application using Angular JS", International Journal of Computer Science and Information Technologies, Vol. 6 (3) , 2015, 2876-2879

[5] https://www.w3schools.com/angular.

[6] https://angularjs.org