e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

ANGULAR JS

Shubham Goyal[*1], Deepanshu Batra[*2]

[*1]Dronacharya College Of Engineering, Gurugram, Haryana, India.

ABSTRACT

AngularJS is a structural framework for dynamic web applications. Angular JS is an open-source JavaScript framework that is used to build web applications. It can be freely used, changed and shared by anyone. Angular Js is developed by Google. It is an excellent framework for building single phase applications and line of business applications. 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 Angular JS is, why Angular JS.

I. INTRODUCTION

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.

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?

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 some reasons why you should be using Angular today.

1. MVC Based :

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.

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.

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. 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

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

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".

4. 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. 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.

III. COMPONENTS OF ANGULAR

Basic Directives :

Directives are nothing but the markers on a DOM element, which tell the compiler to attach a specified event or behaviour to that element.

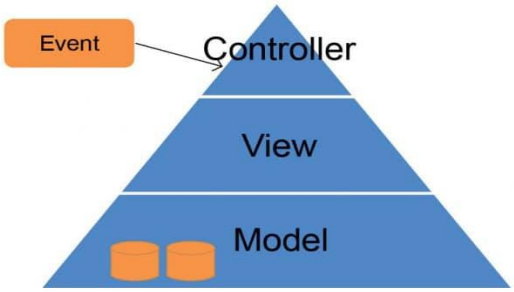
Angular comes with a set of built-in directives.

- **ng-app**: *Initiates an AngularJS application*
- **ng-init**: Initialises application data
- **ng-model***:* Defines the model to be used in AngularJS
- **ng-repeat***:* Repeats HTML elements for each item in a collection
- **ng-bind***:* Binds the data to the element and is an alternative to the interpolation directive
- **ng-show***:* Shows the HTML element
- **ng-hide***:* Hides the model element
- **ng-switch***:* Used to add or remove elements from DOM, based on data
- **ng-if***:* Same as switch, just has a simpler syntax
- **ng-include***:* Includes the HTML elements from other files

MVC architecture :

Model View Controller (MVC) is a software design pattern used for the development of Web applications. It is made up of the following three parts:

- Model – the lowest level, which is responsible for maintaining data.
- View – responsible for displaying all or a portion of the data to the user.
- Controller – controls the interactions between the Model and View.



Model

The model contains the data to be displayed, as well as data to be collected in any input fields or forms. Additionally, it may contain functions that are called based on user input or other activities. A model in Angular is just a plain JavaScript object. It can be a primitive type such as a string, number, Boolean or a complex type such as an object.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

View

View is nothing but how you present the data in a particular format. It may be triggered by the controllers decision. The view in an AngularJS application is created using HTML. View is what users see. It binds data from the model inside your controller, and will get updated automatically if there are model changes from the controller, since Angular has two-way data binding. There is no need to write additional code to achieve this.

Controller

Controller is the place where the application logic resides. It is simply formed by JavaScript classes. In it, we can also call other components to work with. The controller is where the model lives. The controller for a view is used to pull together the model used by the view and handle the input from the consumer of the view. In AngularJS, the controller is resolved by name, from the ng-controller directive.

Dependency Injection :

AngularJS contains a built-in dependency injection mechanism. This is the key to making easily reusable and testable components. It is a software pattern that ensures references to other components are not created directly for the components. Instead of direct instantiation, each component will get references required by other components. So there is no need to find the dependencies for components, as dependencies themselves are configurable. The application will be divided into different components, which can be injected into each other. This makes your application easily reusable, configurable and testable.

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:

1. Value
2. Factory
3. Service
4. Provider
5. 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.

Injecting a value: To inject a value into Angular JS controller function, add a parameter with the same when the value is defined.

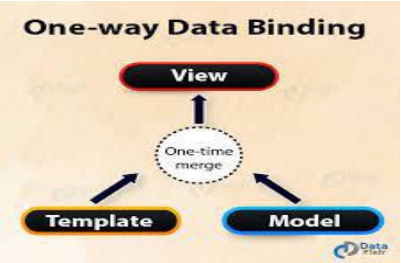
Factory: Factory is a function that is used to return value. When a service or controller needs a value injected from the factory, it creates the value on demand. It normally uses a factory function to calculate and return the value.

Injecting values into factory: To inject a value into Angular JS controller function, add a parameter with the same when the value is defined.

Angular JS follows Two-Way data binding model

- **One-Way Data Binding:**

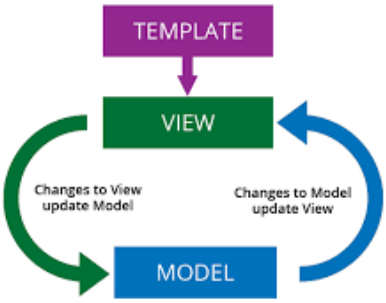
The one-way data binding is an approach where a value is taken from the data model and inserted into an HTML element. There is no way to update model from view. It is used in classical template systems. These systems bind data in only one direction.



e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:04/Issue:06/June-2022 Impact Factor- 6.752 www.irjmets.com

• **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.



IV. CONCLUSION

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 Angular JS is, why Angular JS, MVC Architecture of Angular JS.

V. REFERENCES

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

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

[3] 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.

[4] https://angularjs.org