**Agenda: Introduction**

- What is AngularJS
- MVC Architecture
- Conceptual Overview
- Setting up the Environment
- First Application

What is AngularJS

- History: AJAX → jQuery / Prototype → MVC Frameworks (Knockout / Backbone) → AngularJS
- AngularJS is a very powerful open source and absolutely free **JavaScript library** distributed as **Angular.js** file.
- It's majorly used for developing Single Page Applications (SPA).
- It's based on Model View Controller design pattern.
- It's an extension of HTML DOM and has additional attributes which reduces code and makes it easy to program dynamic and responsive applications.
- Its features include everything we need to build a CRUD app: data-binding, basic templating, directives, form validation, routing, deep-linking, reusable components, dependency injection.
- Angular is built around the belief that **declarative code** is better than imperative when it comes to building UIs and wiring software components together, while imperative code is excellent for expressing business logic.
- In AngularJS, views are pure html pages, and controllers written in JavaScript do the business processing.
- AngularJS applications can run on all major browsers and smart phones, including Android and iOS based phones/tablets.

What is MVC Pattern?
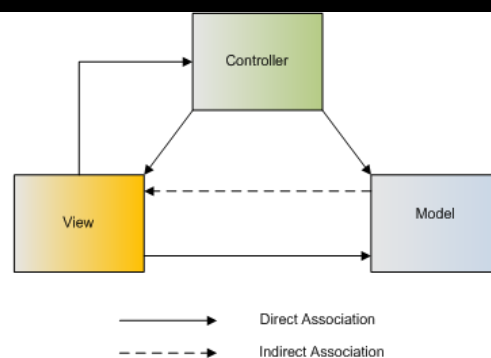
Model: The model is responsible for managing application data. It responds to the request from view and to the instructions from controller to update itself.

View: A presentation of data in a particular format, triggered by the controller's decision to present the data.

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.

AngularJS Conceptual View

Following are the important parts of AngularJS:
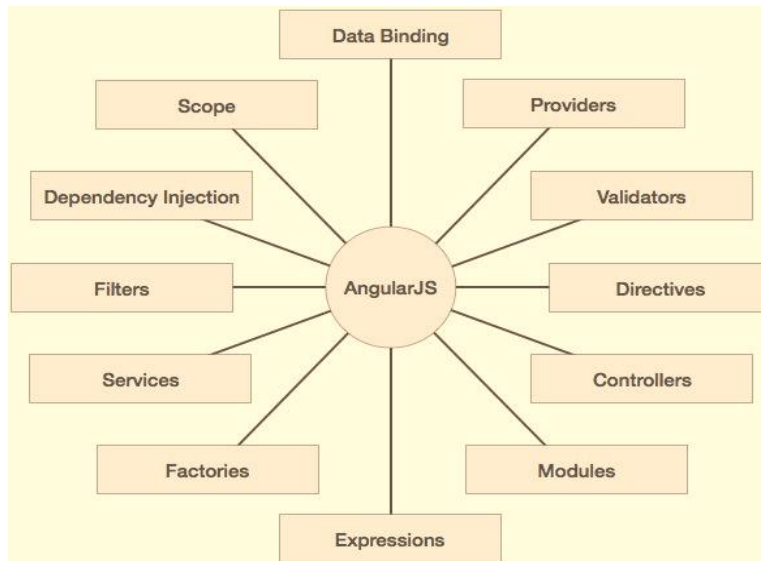
Concept	Description
Template	HTML with additional markup
Directives	Extend HTML with custom attributes and elements
Model	The data shown to the user in the view and with which the user interacts
View	What the user sees (the DOM)
Controller	The business logic behind views
Scope	Context where the model is stored so that controllers, directives and expressions can access it
Expressions	Access variables and functions from the scope
Compiler	Parses the template and instantiates directives and expressions
Filter	Formats the value of an expression for display to the user
Data Binding	Sync data between the model and the view
Dependency Injection	Creates and wires objects and functions
Injector	Dependency injection container
Service	Reusable business logic independent of views
Module	A container for the different parts of an app including controllers, services, filters, directives which configures the Injector

Setting Up of Environment

Visit https://angularjs.org/

Click on Download, in the dialog box choose appropriate option and Download the JS file – Angular.JS or

Angular.min.js.

CDN Link: http://ajax.googleapis.com/ajax/libs/angularjs/1.3.X/angular.min.js

First Application

```html
<!doctype html>
<html >
<head>
   <title>First AngularJS app</title>
   <script src="angular.js"></script>
</head>
<body>
   <div ng-app ng-init="yourFirstName='Sandeep';yourLastName='Soni'">
      <label>First Name:</label>
      <input type="text" ng-model="yourFirstName" placeholder="Enter a first name here">
      <label>Last Name:</label>
      <input type="text" ng-model="yourLastName" placeholder="Enter a last name here">
      <hr>
      <h1>Hello {{yourFirstName + " " + yourLastName}}!</h1>
   </div>
</body>
</html>
```

Understanding the above code:

AngularJS Directives:

The **ng-app** directive defines an AngularJS application.

The **ng-model** directive binds the value of HTML controls (input, select, textarea) to application data.

The **ng-bind** directive binds application data to the HTML view.

The **ng-init** directive initialize AngularJS application variables.