Angular

Description

Angular is a front-end framework. Angular is a platform and framework for building client applications in HTML and Typescript. Angular is written in Typescript. It implements core and optional functionality as a set of Typescript libraries that you import into your apps.

Expectations and Goals

This course will help you master concepts such as Typescript, Dependency Injections, SPA (Single Page Application), Directives, Forms, Pipes, Promises, Observables, and understand the testing of Angular 8 class

Prerequisites

Basics of HTML, CSS and Java-script.

Course Schedule

Module	Topic
Module 1	Introduction to Angular What is Angular? Angular 5 v/s 4 v/s 2 v/s AngularJS Angular CLI NodeJS Introduction (NPM) Setup of NodeJs and Angular What is Typescript? How does Angular get started? First Angular App
Module 2	Components Overview Introduction to Components Creating components Role of AppModule & Component Declaration Registering Components Using Registered Components Creating Components with CLI Multiple components & passing data Nesting Components Working with Component templates Working with Component Styles Understanding Component Selector
Module 3	Components & Data binding Introduction to Modules & Data binding Splitting Apps into Components Property & Event binding overview Binding to Custom Properties Assigning an Alias to Custom Properties Binding to Custom Events Assigning an Alias to Custom events Custom Property and Event Binding Summary Understanding View Encapsulation Using Local References in Templates Getting Access to the Template & DOM with @ViewChild Projecting Content into Components with ng-content Understanding the Component Life cycle.

Module 4	**Understanding Directives** Using ngIf to Output Data Conditionally Enhancing ngIf with an Else Condition Output Lists with ngFor Styling Elements Dynamically with ngStyle Applying CSS Classes Dynamically with ngClass Creating Basic Attribute Directive Using the Renderer to build Better Attribute Directive More about Renderer Listen to Host Events using HostListener Bind to Host Properties using HostBinding Binding to Directive Properties Behind the scenes of Structural Directives What is ngSwitch?
Module 5	**Services & Dependency Injection** Introduction to Dependency Injection Why do we need Services ? Creating a Logging Service Injecting the Logging Service into Components Creating a Data Service Understanding Hierarchical Injector How many Instances of Service? Injecting Services into Services Using Service for Cross-Component Communication
Module 6	**Changing Pages with Routing** What is Routing? Why do we need a Router? Setting up and Loading Routes Navigating with Router Links Understanding Navigation Paths Styling Active Router Links Navigating Programmatically Using Relative Paths in Programmatic Navigation Passing Parameters to Routes Fetching Route Parameters Fetching Route Parameters Reactively Route Observables Passing Query Paramters and Fragments Retrieving Query Parameters & Fragments Setting up Child(Nested) Routes Configuring the Handling of Query Parameters Redirection & Wildcard Routes Outsourcing the Route Configuration Introduction to Route Guards Protecting Routes with canActivate Controlling Navigation with canDeactivate Passing static data to a Route Resolving Dynamic Data with the resolve Guard Understanding Location strategies Understanding Observables

Module 7	**Template Driven Forms** Introduction to handling forms Why do we need Angular's help? Template Driven(TD) v/s Reactive Approach Creating Template driven Forms & Registering Controls Submitting & Using the Form Understanding Form State Accessing the Form with @ViewChild Adding Validation to check User Input Built-in Validators & Using HTML5 Validation Using the Form State Outputting Validation Error Messages Set Default Values with ngModel Property Binding Using ngModel with Two-Way-Binding Grouping Form Controls Handling Radio Buttons Setting & Patching form values Using Template Driven Form Data Resetting Template Driven Forms
Module 8	**Reactive Forms** Introduction to Reactive Approach Creating a Reactive Form in Code Syncing HTML and Form Submitting Reactive Forms Adding Validation to Reactive Forms Getting Access to Controls Grouping Controls Arrays of Form Controls Creating Custom Validators Using Error Codes with Reative Forms Creating Custom Async Validator Reacting to Status or Value Changes
Module 9	**Transport Output using Pipes** Introduction to Pipes Why are Pipes useful? Using Pipes Parameterizing Pipes Chaining Multiple Pipes Creating a Custom Pipe Parameterizing a Custom Pipe Creating a Filter Pipe
Module 10	**Making HTTP Requests** Introduction to Http Requests How HttpRequests Work in SPAs Sending Requests Adjusting Request Headers Sending GET Requests Sending a PUT Request

Module 11	**Angular Modules & Optimizing Apps** The idea behind Modules Understanding App Module Understanding Feature Modules Creating a Feature Module Registering Routes in Feature Modules Understanding Shared Modules Creating a Shared Module Creating the Auth Feature module Understanding Lazy Loading
Module 12	**HttpClient** Introduction to HttpClient Unlocking the HttpClient Request Configuration & Response Requesting Events Setting Headers Http Parameters Progress Interceptors Modifying Requests in Interceptors Multiple interceptors
Module 13	Angular Animations Introduction to Angular Animations Animation Triggers & State Switching between States Transitions Advanced Transitions Transition Phases The "void" State
Module 14	Project work and documentation