A javascript framework for developing SPA.

Developed by Google

Uses typescript/javascript/DART

**Features**

Modularity - can use individual modules and can combine to develop large Apps

Easy

Dependency Injection support

Bindings

Routing

REST Api Support

Event handling

Template Support

Angular CLI used to create templates and components

Angular Supports many IDE like Visual studio and so on

Tools – Angular CLI , IDE

Why Angular

* React is not dedicated for SPA, Uses lots of Third Party tools , not suitable for large apps ,
* As angular provide many built in features for routing , binding , dependency injection and etc

Requirements - NodeJS, TypeScript , Angular CLI

Npm install –g @angular/cli

Once intall the angularCLI we wil get a command ng command

Ng new ,Ng generate

Angular1 is purely uses Javascript , uses MVVM patter , and contains scopes , filters, controller, Directives

**Angular2**

Components(Replacement of scope and controllers)

Pipes (Replacement of filters )

Services

Directives

***Architecture***

Module is the main building block ,

Application is combination of different modules

Module = collection of classes . (its package or namespace )

**An application can have more than one module**

Must have only one main module

**Angular Module**

**Component**

Code{}

Html<>

**Pipes**

|currency

**Services**

Class data{

}

**Directives**

Code{}

Other Modules

Built in , Routing,..

**AppModule :**

---------------------------------------------

@ngModule (

Declaration:[AppComponent,HomeComponent, AboutComponent,.. Components, pipes, directives ],

Providers :[MovieService,UserService, Services],

Imports:[UserModule,BuiltinModules,BrowserModule, all the modules],

Exports :[ModuleNames to be exported ,..],

entryComponets :[dynamic Components ]

Bootstrap:[AppComponent]// setting the startup

)

Export class AppModule {

}

Let us consider in a project which is having first, second , third Module

@ngModule for FirstModule

Declarations – Compoenenets for First Module

Imports – secondmodule , third Module , other module

Exports – Exports the modules to other modules which is consuming appmodule

Importing some module , adding some more functionality and export

Convention

Module names should ends with modules

Service names should ends with service

There must be one Root Component , generally it will be with Name APP

Main module should contain Root Component

Bootstrap Sequence

Index.html - <app></app> -🡪 main.ts 🡪 appModule 🡪 finds app and display

**Component**

@component({

Selector:home,

Template:`html code tags`, // template is used when we want render some inline html

Templateurl:’/htmlpageurl.html’, // hmtl file url

Styles:`[ all styles can be written ]`,

StylesUrls:[styleFilename.css,home.css,..All the css file required ]

})

Export class Home{

Constructor(){}

doClick(){}

}

On home page u can have

<div>

<button id=”btn” (click)=’doClick’ />

</div>

* Angular Events ie click must be in ().

**Ex: to Create a new component**

Create a new file in app folder >> name it as sample.component.ts

import {Component} from '@angular/core'

@Component({

selector:'sample-comp',

template:'<h1>Sample Component</h1>',

styles:["h1{color:red}"]

})

export class SampleComponent{

title='Sample Component'

}

**Now at appModule.ts we need to register the component . write the following**

import {SampleComponent } from './sample.component'

@NgModule({

declarations: [

AppComponent,SampleComponent

],

})

**To Dispaly the sample component we need to use the selector**

At appcomponent.html include the selector tag for samplecomponent

<div style="text-align:center">

<h1>

Welcome to {{ title }}!

</h1>

</div>

<h2>Here are some links to help you start: </h2>

<ul>

</ul>

**<sample-comp></sample-comp>**