**What is Angular?**

Angular is a client side framework which is used to create Publications the Framework provides skeleton of the product and specific guidelines where to write which type of code.

1. It can be used in combination with any service at platforms like Java, NodeJS, asp.net, PHP, Python.
2. it is developed using typescript language
3. it is most used to client side framework
4. it is developed by Google
5. it is free to use
6. it is open source
7. it is cross platform
8. it is cross browser compatible
9. it is mainly used to create data bindings where we can separate prepare application logic & DOM manipulations

Goals of Angular

1. separation of demand equation from application logic
2. make single page application development easiest
3. separation of business logic from application logic
4. enable unit testing the components can be unit tested individually

Versions of angular:

Angular 2

Angular 4

Angular 5

Angular 6

Angular 7

AngularJs Vs Angular

Building Blocks Of Angular

1. Component : application state and application logic
2. Metadata details about the component or model
3. Template design HTML logic
4. Data Binding connection between HTML elements and component property
5. Module group of components, directives and pipes
6. Services reusable code for business project
7. Dependency Injection injecting service objects into component
8. Directive manipulating Dom elements
9. Pipe transforming values before display

**Angular Architecture:**

**Steps to prepare First Angular Application:**

1. Installing node JS
2. Installing typescript
3. Installing git
4. Installing Visual Studio code
5. Creating application folder
6. Install @angular/cli package
7. Creating new angular application
8. Open the application in Visual Studio code
9. Edit code in in app.component.html file
10. Compile the application
11. Run the application

**Angular Application Directory Structure:**

**1.package.json:**

* Name represent name of the application
* Version represent version of the application
* License present license of the application

MIT stands for Massachusetts institute of technology

* It allows to create private applications that can be used either privately within organization and also can be shared with other known organizations

ISC stands for internet system consortium

* It allowed to create public applications that can be used anywhere
* Scripts represent a set of commands that can run on the command from to run test the application using command
* Private represent weather the application should be used privately within the same organization or not
* Dependencies represent the list of packages that are to be installed to run the application

These packages will be installed in both the developer edition and production server

* Devdependencies Represents the list of packages that are to be installed to develop the application

These packages will be installed only in the developer mission not in the production server

PACKAGES OF ANGULAR

* Angular Framework is a collection packages
* Each feature of the Framework is represented as a package
* For example routing is available as a package called @ angular / router
* We have to install those packages in order to develop angular application
* The packages of angular are divided into two types

1. angular packages and 2.non angular packages

1. Angular packages these packages are part of the angular 2 + framework

1. @ angular / core provides classes & interfaces that are related to decorators, component life cycle, dependency injection(@Component,@NgModule,@Input,@Inject,@Injectable,..)

2. @ angular / common provides common directives(ngIf,ngFor,ngSwitch,ngClass,ngStyle,..) & pipes(uppercase,lowercase,date,..)

3. @ angular / platform – browser provides runt time services (error handling, history handling) that are needed while running application in the browser.

4. @ angular / compiler used to compile the template to javascript code

5. @ angular /platform-browser-dynamic used to bootstrap(start) executing a module, which execution should be started automatically at application startup.

6. @ angular / forms used for creating two way data binding and validations

this package has two models

1.FormsModule and ReactiveFormsModule

7. @ angular /router used to creating routing

this package has one model that is RouterModule

8. @ angular /http used to send ajax request to server and get ajax response from the server

this package has one module

HttpClientModule

9. @ angular / animation

used to create animations

has one module AnimationsModule

10. @ angular/material

used to use angular material design

11. @ angular/cli this package provide a set of commands to create new angular application and its code items such as

Components, pipes, directives, services, etc

2. Non angular packages

These packages are not part of angular framework but provide the third party companies and needed in angular application

1. typescript call typescript compiler which is used to compile typescript files into JavaScript files

2. system.JS used to load both angular Framework created and application program related dot JS files into the browser

3. core-js this package contains policies which are needed to run angular 5 application in internet explorer 9

4. Rx Js

this package reactive extensions for JavaScript provides necessary code for making ajax calls to server

5.Zone.js

this package identify the events in the browser and inform the same to angular framework so that it can perform change detection

6. Jasmine

used to write test cases for unit testing of components

7.karma

this package is used to execute test cases on different browsers

8.tslint

this package used to check whether the typescript files are following set of rules or not

**2. tsconfig.json**

Every compiler has some configuration settings

This file used to set configurations settings of the TSC

The TSC compiler automatically reads the tsconfig.json file and then only it compiles the .ts files into .js files.

**3. tslint.json**

this file contains configuration settings for tslint tool which is used to verify whether the typescript files are following a set fo coding standards or not

4. protractor.conf.js

this file contains configuration settings for protractor tool which is used to perform unit testing of components

the protractor tool used to execute the test cases that are defined using Jasmine

5. karma.conf.js

this file contains configuration settings for Karma tool which is used to execute unit test cases on multiple browsers

6. angular.cli.json

this file contains configuration settings for CLI tool used to create, compile and run the application

it contains setting such as homepage that is index.html, startup filename that is main.ts, CSS file name styles.css

7. polyfills.ts

this file contains configuration settings for importing polyfills which are needed to run angular applications on old browsers such as Internet Explorer

8. src/styles.css

this file contains CSS styles that are applicable for entire application

9. src/index.html

this file is a homepage for the entire application

the content of the entire application appears in the same HTML file only

this file invokes the AppComponent using <app-root> tag.

10. src/main.ts

this is the first types of file that is executed in angular application

11. src/app.app.module.ts

this file contains definition of AppModel

angular application can has any number of modules

it should contain at least one module that is called as AppModule

this file imports AppComponent from app.component.ts file and bootstrap the same in AppModule

12. src/app/app.component.ts

this file contains definition of AppComponent

angular application can has any number of components

it should contain at least one module that is called as AppComponent

13. src/app/app.component.html

This file contains actual content of component

Every component should have a template

This template content will be rendered into <app-root> tag in index.html

14. src/app/app.component.css

This file contains css styles of AppComponent

15. src/app/app.component.spec.ts

This file contains test cases for AppComponent

**Application Modes In Angular**

1. **Development mode**

Change detection occurs twice. Raises error if any difference detected between first attempt and second attempt

1. **Production mode**

Change detection occurs only once

Startup module

Angular application can has any number of modules

The startup module is a module which needs to be executed first in the angular application

By default startup module name is AppModel

Loading the startup model is also called as bootstrapping the module

1. **Component:**

* The component class represent certain section of the webpage for example login form is represented as a login component
* The component class includes properties(to store data) & methods(event handler methods to manipulate data)
* Every angular application contain at least one component which is called as AppComponent.
* We can create any number of components in this project
* The component is invoke through a custom tag
* For example login component is input through <login> tag
* The custom type is also called as selector
* The component class should have a decorator called @Component to define that the class is in component class

Syntax:

import {Component } from “@angular/core”;

@Component(metadata)

Class ClassName

{

Property:datatype=value;

Method(arg):return type

{  
}

}

Meta Data Properties of Component:

1. Selector

Present collector to know the component

1. Template

Represent the template content of the component

1. Template url

Represent the html file that has to be rendered in the component is in

1. Style url

Represent the list of stylesheet that have to be loaded for the component

1. Providers

Represents the list of services to be imported into the component

1. Animations

Represent the list of animations to be performed in the component

1. **Module :**

* Module is a part of the project
* Module is a collection of components, directives and pipes that are related to one specific tasks of the project

Example

Net banking project contains models like saving account module, credit card module

* Every angular application should contain at least one module which is called as root module or app module
* The AppComponent will be a part of the AppModule
* Modules in share its components and pipes to other modules
* Module is a class with @NgModule decorator

Syntax:

Import {NgModule} from “@angular/core”;

@NgModule(metada)

Class ClassName

{

}

Meta Data Properties of Component:

1. Declarations

In present the list of components and pipes that are members of the current model

1. Imports

Represent the list of modules that you want to import in the current module

You must input BrowserModule into the browser which can be imported @angular/platform-browser

1. Exports

The present the list of components are pipes that are to be exported to other modules

1. Bootstrap

Represent the component that is to be displayed in the webpage

Only AppModule has to bootstrap “app component”

Other modules should not bootstrap any component

1. Providers

Represent list of services to be imported into the module

**DATA BINDING**

Data binding is a relation between component and template

When the value of component is changed the template will be changed automatically

When the value of template is changed the component will be changed automatically

Data binding is four types

1. Interpolation binding

2. Property binding

3. Event binding

4. Two way binding