ANGULAR 6

Saramma Varghese

OVERVIEW

- Angular 6 released in May 2018
- The structure of Angular is based on the components/services architecture.
- features added to Angular 6 −
  - Updated Angular CLI, Command Line interface
  - Updated CDK, Component Development Kit
  - Updated Angular Material
  - Usage of RxJS, a reactive JS library
  - Angular Element

ENVIRONMENTAL SETUP

- To install Angular 6, we require the following −
  - Nodejs
  - Npm
  - Angular CLI
  - IDE for writing your code
- To see the version of nodejs currently installed on your system , type **node –v** in terminal

CONT...

- To check if npm is installed, type **npm -v** in the terminal.
- To install angular cli on your system, Type **npm install -g @angular/cli**

COMPONENTS

- Components are basically classes that interact with the .html file of the component.
- The file structure has the app component-
  - **app.component.css**
  - **app.component.html**
  - **app.component.spec.ts**
  - **app.component.ts**
  - **app.module.ts**

CONT...

- command to create the component-

  ng generate component new-cmp

MODULE

- Is the place where you can group the components, directives, pipes, and services, which are related to the application.
- To define module, we can use the **NgModule**.
- The NgModule needs to be imported as-

  import { NgModule } from '@angular/core‘
- It starts with **@NgModule** and contains an object which has declarations, import s, providers and bootstrap.

CONT...

- Structure of NgModule –

```
@NgModule({
   declarations: [
      AppComponent
   ],
   imports: [
      BrowserModule
   ],
   providers: [],
   bootstrap: [AppComponent]
})
```

- Declaration : It is an array of components created.

CONT...

- Import : It is an array of modules required to be used in the application.
- Providers :This will include the services created.
- Bootstrap :This includes the main app component for starting the execution.

DATA BINDING

- Use curly braces for data binding - **{{}}**; this process is called interpolation.
- Eg ,
  - In .ts file :
    - title = 'Angular 4 Project!';
  - In .html file :
    - <h1> Welcome to {{title}} </h1>

EVENT BINDING

- A user interacts with an application (keyboard movement, a mouse click, or a mouseover),it generates an event.
- These events need to be handled to perform some kind of action.
- Eg ,
  - In .html file

```
<button (click)=“clickFunction($event)">
  Click Me
</button>
```

CONT...

- In .ts file

```
clickFunction(event) {
  console.log(event);
 }
```

TEMPLATES

- Tag used is **<ng-template>**
- Eg ;

```
<div>
 <span *ngIf = "isavailable ; then condition1 </span> <ng-template
    #condition1>
        Condition is valid
    </ng-template>
</div>
```

DIRECTIVES

- declared as **@directive.**
- Different directives :
  - **Component Directives**: deals how the component should be processed, instantiated and used at runtime.
  - **Structural Directives** : deals with manipulating the dom elements. For example, ***ngIf** and ***ngFor**.
  - **Attribute Directives** :deal with changing the look and behavior of the dom element.

CONT...

```
import { Directive } from '@angular/core';
@Directive({
   selector: '[appChangeText]'
})
export class ChangeTextDirective {
   constructor() { }
}
```

PIPES

- Pipes were earlier called filters .
- The | character is used to transform data.
- syntax for the same -

  {{ Welcome to Angular 6 | lowercase}}
- Angular 6 provides some built-in pipes −
  - Lowercasepipe
  - Uppercasepipe
  - Datepipe
  - Currencypipe
  - Jsonpipe

CONT...

- Percentpipe
- Decimalpipe
- Slicepipe

- Eg ;
  - In .ts file

    title = “ example fore pipe”
  - In .html file

    {{title | uppercase}}

ROUTING

- Means navigating between pages.
- have to include the router module-

  import { RouterModule} from '@angular/router';
- RouterModule refers to the **forRoot** which takes an input as an array, which in turn has the object of the path and the component
- Path is the name of the router and component is the name of the class**.**

SERVICES

- There is situation where we need some code to be used everywhere on the page.
- It can be for data connection .
- Command for creating service –

  ng g service myservice

- In app.module.ts, the created service should added to providers.

CONT...

- Eg ;
  - In service file

    showTodayDate() { // function created

    let ndate = new Date();

    return ndate;

    }
  - In .html file

    this.todaydate = this.service.showTodayDate(); // object of service file created in constructor

HTTP CLIENT

- It will help us fetch external data, post to it, etc.
- We need to import http to make use of the service,

import { HttpClient } from '@angular/common/http';

CONT...

```
import { Component } from '@angular/core';
import { HttpClient } from '@angular/common/http';
@Component({
   selector: 'app-root',
   templateUrl: './app.component.html',
   styleUrls: ['./app.component.css']
})
export class AppComponent {
   constructor(private http: HttpClient) { }
   httpdata;
   ngOnInit() {
      this.http.get("http://jsonplaceholder.typicode.com/users")
      .subscribe((data) => this.displaydata(data));
   }
   displaydata(data) {this.httpdata = data;}
}
```

FORMS

- 2 different forms-
  - template driven form
  - model driven forms
- Template Driven Form : most of the work is done in the template.

```
<form #userlogin = "ngForm" (ngSubmit) = "onClickSubmit(userlogin.value)" >
   <input type = "text" name = "emailid" placeholder = "emailid" ngModel>
   <br/>
   <input type = "password" name = "passwd" placeholder = "passwd" ngModel>
   <br/>
   <input type = "submit" value = "submit">
</form>
```

CONT...

- Model Driven Form : most of the work is done in the component class.

```
<div>
   <form [formGroup] = "formdata" (ngSubmit) = "onClickSubmit(formdata.value)" >
      <input type = "text" class = "fortextbox" name = "emailid" placeholder = "er
         formControlName="emailid">
      <br/>

      <input type = "password" class = "fortextbox" name="passwd"
         placeholder = "passwd" formControlName = "passwd">
      <br/>

      <input type = "submit" class = "forsubmit" value = "Log In">
   </form>
</div>
<p>
   Email entered is : {{emailid}}
</p>
```

CLI

- Angular CLI makes it easy to start with any Angular project.
- To work with Angular CLI, we need to install –

  npm install -g @angular/cli

- To create a new project-

  ng new PROJECT-NAME

Component	ng g component new-component
Directive	ng g directive new-directive
Pipe	ng g pipe new-pipe
Service	ng g service new-service
Module	ng g module my-module

THANK YOU...